

Zinstruct API

Version 1.0.0 · <https://app.zinstruct.com>

Read your school's data from your own tools.

Authenticating

Every request carries an API key as a bearer token:

```
Authorization: Bearer YOUR_API_KEY
```

Create a key at **Settings → API keys**. The full key is shown once, when you create it. Keys are read-only, reach only your own school, and stop working the moment you revoke them.

Start with `GET /api/reporting/v1/health` — it proves the key and the header are right before you touch any data.

Conventions

- Field names are `snake_case` ; timestamps are ISO-8601 in UTC; money is always **integer cents**, never a decimal.
- List endpoints return `{ data, total, page, page_size }`. `total` counts every matching row, not just the page.
- Paginate with `?page=` and `?page_size=`. `page_size` is clamped to 1–100, and defaults to 50.
- Errors return `{ "error": { "code", "message" } }` with a matching HTTP status. Branch on `code` — it is stable; `message` may be reworded.

Rate limits

Requests are limited per key: **600 requests per 60 seconds**. Over the limit you get `429` with code `rate_limited`. Give each integration its own key so one busy tool cannot exhaust another's budget.

Endpoints at a glance

<code>GET /api/public/v1/contacts</code>	<code>GET /api/reporting/v1/courses/{course_id}/modules</code>
<code>GET /api/public/v1/contacts/{id}</code>	<code>GET /api/reporting/v1/students</code>
<code>GET /api/public/v1/products</code>	<code>GET /api/reporting/v1/students/{student_id}</code>
<code>GET /api/public/v1/products/{id}</code>	<code>GET /api/reporting/v1/students/{student_id}/watch-history</code>
<code>GET /api/public/v1/offers</code>	<code>GET /api/reporting/v1/lessons</code>
<code>GET /api/public/v1/offers/{id}</code>	<code>GET /api/reporting/v1/lessons/{lesson_id}/retention</code>
<code>GET /api/public/v1/enrollments</code>	<code>GET /api/reporting/v1/summary</code>
<code>GET /api/public/v1/enrollments/{id}</code>	<code>GET /api/reporting/v1/health</code>
<code>GET /api/reporting/v1/courses</code>	

Core data

GET /api/public/v1/contacts

List contacts

Newest first. Combine filters to narrow the list; omitted filters are not applied.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1-100 range.
role	query	string	Only contacts with this role.
status	query	string	Only contacts with this `contact_status`, e.g. `active`.
marketing_status	query	string	Only contacts with this email consent state, e.g. `subscribed`.

RESPONSE FIELDS — CONTACT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
email	string	Email address.
name	string null	Display name. Falls back to first + last name when the name field is empty.
first_name	string null	Given name.
last_name	string null	Family name.
role	string null	Your own classification of this contact.
phone	string null	Phone number. Falls back to the mobile number when the primary is empty.
contact_status	string	Lifecycle state, e.g. `active` or `archived`. Defaults to `active`.
marketing_status	string	Email consent, e.g. `subscribed` or `unsubscribed`. Defaults to `unsubscribed`.
source	string null	Where the contact first came from. Immutable first-touch provenance.
lifetime_value_cents	integer	Total revenue attributed to this contact, in cents.
created_at	string null	When the contact was created. ISO-8601, UTC.
last_activity_at	string null	Most recent recorded activity. ISO-8601, UTC.

RESPONSES

- 200 — A page of contacts.
- 401 — The token is missing, malformed, invalid, or revoked.

- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/contacts?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/contacts?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/contacts?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/contacts?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/contacts?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)
```

```
puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/contacts?
page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}
```

GET /api/public/v1/contacts/{id}

Get one contact

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
id *	path	string (uuid)	The contact id.

RESPONSE FIELDS — CONTACT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
email	string	Email address.
name	string null	Display name. Falls back to first + last name when the name field is empty.
first_name	string null	Given name.
last_name	string null	Family name.
role	string null	Your own classification of this contact.
phone	string null	Phone number. Falls back to the mobile number when the primary is empty.
contact_status	string	Lifecycle state, e.g. `active` or `archived`. Defaults to `active`.
marketing_status	string	Email consent, e.g. `subscribed` or `unsubscribed`. Defaults to `unsubscribed`.
source	string null	Where the contact first came from. Immutable first-touch provenance.
lifetime_value_cents	integer	Total revenue attributed to this contact, in cents.
created_at	string null	When the contact was created. ISO-8601, UTC.
last_activity_at	string null	Most recent recorded activity. ISO-8601, UTC.

RESPONSES

- **200** — The contact.
- **401** — The token is missing, malformed, invalid, or revoked.
- **404** — No contact with that id.
- **429** — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/contacts/00000000-0000-0000-0000-000000000000' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/contacts/00000000-0000-0000-0000-000000000000', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
```

```

}))

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)

```

PYTHON

```

import requests

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/contacts/00000000-0000-0000-0000-000000000000",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/contacts/00000000-0000-0000-0000-000000000000');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/contacts/00000000-0000-0000-0000-000000000000')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"

```

```

)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/contacts/00000000-0000-0000-0000-000000000000", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}

```

GET /api/public/v1/products

List products

Newest first.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1–100 range.
type	query	string	Only products of this type, e.g. `course`.
status	query	string	Only products with this status, e.g. `published`.

RESPONSE FIELDS — PRODUCT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
type	string	Product type, e.g. `course`, `coaching`, `community`, `download`, `newsletter`, `podcast`.
sub_type	string null	Further classification within the type, where one applies.
title	string	Product title.
description	string null	Product description.
status	string	`draft` or `published`. Defaults to `draft`.
created_at	string null	When the product was created. ISO-8601, UTC.
updated_at	string null	When the product was last changed. ISO-8601, UTC.

RESPONSES

- **200** — A page of products.
- **401** — The token is missing, malformed, invalid, or revoked.
- **429** — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/products?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/products?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}
```

```
const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/products?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/products?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/products?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/products?
page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
```

```
    if err != nil {  
        panic(err)  
    }  
    defer res.Body.Close()  
  
    if res.StatusCode >= 400 {  
        panic(fmt.Sprintf("API returned %d", res.StatusCode))  
    }  
  
    var body map[string]any  
    json.NewDecoder(res.Body).Decode(&body)  
    fmt.Println(body)  
}
```

GET /api/public/v1/products/{id}

Get one product

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
id *	path	string (uuid)	The product id.

RESPONSE FIELDS — PRODUCT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
type	string	Product type, e.g. `course`, `coaching`, `community`, `download`, `newsletter`, `podcast`.
sub_type	string null	Further classification within the type, where one applies.
title	string	Product title.
description	string null	Product description.
status	string	`draft` or `published`. Defaults to `draft`.
created_at	string null	When the product was created. ISO-8601, UTC.
updated_at	string null	When the product was last changed. ISO-8601, UTC.

RESPONSES

- 200 — The product.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No product with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/products/00000000-0000-0000-0000-000000000000' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/products/00000000-0000-0000-0000-000000000000', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests
```

```

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/products/00000000-0000-0000-0000-000000000000",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/products/00000000-0000-0000-0000-000000000000');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/products/00000000-0000-0000-0000-000000000000')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/products/00000000-0000-0000-0000-000000000000", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {

```

```
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}
```

GET /api/public/v1/offers

List offers

Newest first.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1–100 range.
product_id	query	string	Only offers selling this product.
status	query	string	Only offers with this status, e.g. `published`.

RESPONSE FIELDS — OFFER

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
product_id	string (uuid)	The product this offer sells.
name	string	Offer name.
description	string null	Offer description.
price_cents	integer	Price in the smallest currency unit. Money is always integer cents — never a float.
currency	string	ISO-4217 currency code, lowercase. Defaults to `usd`.
plan	string	`one_time` or a recurring plan. Defaults to `one_time`.
interval	string null	Billing interval for recurring plans, e.g. `month` or `year`. Null for one-time offers.
interval_count	integer	How many intervals per billing cycle.
status	string	`draft` or `published`. Defaults to `draft`.
pwYW	boolean	True when the buyer names their own price.
trial_days	integer null	Free trial length in days, or null for no trial.
created_at	string null	When the offer was created. ISO-8601, UTC.
updated_at	string null	When the offer was last changed. ISO-8601, UTC.

RESPONSES

- **200** — A page of offers.
- **401** — The token is missing, malformed, invalid, or revoked.
- **429** — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/offers?page=1&page_size=50' \  
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/offers?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/offers?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/offers?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/offers?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main
```

```

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/offers?
page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}

```

GET /api/public/v1/offers/{id}

Get one offer

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
id *	path	string (uuid)	The offer id.

RESPONSE FIELDS — OFFER

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
product_id	string (uuid)	The product this offer sells.
name	string	Offer name.
description	string null	Offer description.
price_cents	integer	Price in the smallest currency unit. Money is always integer cents — never a float.
currency	string	ISO-4217 currency code, lowercase. Defaults to `usd`.
plan	string	`one_time` or a recurring plan. Defaults to `one_time`.
interval	string null	Billing interval for recurring plans, e.g. `month` or `year`. Null for one-time offers.
interval_count	integer	How many intervals per billing cycle.
status	string	`draft` or `published`. Defaults to `draft`.
pwYW	boolean	True when the buyer names their own price.
trial_days	integer null	Free trial length in days, or null for no trial.
created_at	string null	When the offer was created. ISO-8601, UTC.
updated_at	string null	When the offer was last changed. ISO-8601, UTC.

RESPONSES

- 200 — The offer.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No offer with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/offers/00000000-0000-0000-0000-000000000000' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/offers/00000000-0000-0000-0000-000000000000', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})
```

```

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)

```

PYTHON

```

import requests

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/offers/00000000-0000-0000-0000-000000000000",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/offers/00000000-0000-0000-0000-000000000000');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/offers/00000000-0000-0000-0000-000000000000')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {

```

```
req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/offers/00000000-0000-0000-0000-000000000000", nil)
req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

res, err := http.DefaultClient.Do(req)
if err != nil {
    panic(err)
}
defer res.Body.Close()

if res.StatusCode >= 400 {
    panic(fmt.Sprintf("API returned %d", res.StatusCode))
}

var body map[string]any
json.NewDecoder(res.Body).Decode(&body)
fmt.Println(body)
}
```

GET /api/public/v1/enrollments

List enrollments

Most recently enrolled first.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1–100 range.
product_id	query	string	Only enrollments in this product.
contact_id	query	string	Only enrollments belonging to this contact.
status	query	string	Only enrollments with this status, e.g. `active`.

RESPONSE FIELDS — ENROLLMENT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
contact_id	string (uuid)	The enrolled contact.
product_id	string (uuid)	The product they have access to.
granted_via_offer_id	string null	The offer that granted access, when access came from a purchase or claim.
status	string	Access state, e.g. `active`. Defaults to `active`.
progress_pct	number	Completion percentage, 0–100.
enrolled_at	string null	When access began. ISO-8601, UTC.
completed_at	string null	When the product was completed, or null. ISO-8601, UTC.

RESPONSES

- **200** — A page of enrollments.
- **401** — The token is missing, malformed, invalid, or revoked.
- **429** — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/enrollments?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/enrollments?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}
```

```
const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/enrollments?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/enrollments?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/enrollments?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/enrollments?page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")
```

```
res, err := http.DefaultClient.Do(req)
if err != nil {
    panic(err)
}
defer res.Body.Close()

if res.StatusCode >= 400 {
    panic(fmt.Sprintf("API returned %d", res.StatusCode))
}

var body map[string]any
json.NewDecoder(res.Body).Decode(&body)
fmt.Println(body)
}
```

GET /api/public/v1/enrollments/{id}

Get one enrollment

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
id *	path	string (uuid)	The enrollment id.

RESPONSE FIELDS — ENROLLMENT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	Stable identifier.
contact_id	string (uuid)	The enrolled contact.
product_id	string (uuid)	The product they have access to.
granted_via_offer_id	string null	The offer that granted access, when access came from a purchase or claim.
status	string	Access state, e.g. `active`. Defaults to `active`.
progress_pct	number	Completion percentage, 0–100.
enrolled_at	string null	When access began. ISO-8601, UTC.
completed_at	string null	When the product was completed, or null. ISO-8601, UTC.

RESPONSES

- 200 — The enrollment.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No enrollment with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/public/v1/enrollments/00000000-0000-0000-0000-000000000000' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/public/v1/enrollments/00000000-0000-0000-0000-000000000000', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests
```

```

res = requests.get(
    "https://app.zinstruct.com/api/public/v1/enrollments/00000000-0000-0000-0000-000000000000",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/public/v1/enrollments/00000000-0000-0000-0000-000000000000');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/public/v1/enrollments/00000000-0000-0000-0000-000000000000')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/public/v1/enrollments/00000000-0000-0000-0000-000000000000", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {

```

```
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}
```

Engagement

GET /api/reporting/v1/courses

List courses with engagement

Every course with its structure counts, enrollment count, watch activity, and mean completion. Watch metrics read a materialized table and stay at 0 until viewing is recorded.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1–100 range.

RESPONSE FIELDS — COURSE

FIELD	TYPE	DESCRIPTION
id	string (uuid)	The course product id.
title	string	Course title.
modules_count	integer	Modules in the course.
lessons_count	integer	Lessons across all modules.
students_enrolled	integer	Contacts with an enrollment in this course.
lessons_watched_over_30s	integer	Lesson views exceeding 30 seconds of watch time.
distinct_students_watched	integer	Distinct students with any recorded watch time.
avg_completion_pct	number	Mean enrollment progress, 0–100.

RESPONSES

- 200 — A page of courses.
- 401 — The token is missing, malformed, invalid, or revoked.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/courses?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/courses?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}
```

```
const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/courses?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/courses?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/courses?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/courses?page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
```

```
    if err != nil {  
        panic(err)  
    }  
    defer res.Body.Close()  
  
    if res.StatusCode >= 400 {  
        panic(fmt.Sprintf("API returned %d", res.StatusCode))  
    }  
  
    var body map[string]any  
    json.NewDecoder(res.Body).Decode(&body)  
    fmt.Println(body)  
}
```

GET /api/reporting/v1/courses/{course_id}/modules

List a course's modules with engagement

Modules in their display order, each with lesson count and watch activity.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
course_id *	path	string (uuid)	The course product id.
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1-100 range.

RESPONSE FIELDS — COURSEMODULE

FIELD	TYPE	DESCRIPTION
id	string (uuid)	The module id.
title	string	Module title.
lessons_count	integer	Lessons in this module.
lessons_watched_over_30s	integer	Lesson views exceeding 30 seconds of watch time.
distinct_students_watched	integer	Distinct students with any recorded watch time in this module.

RESPONSES

- 200 — A page of modules.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No course with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/courses/00000000-0000-0000-0000-000000000000/modules?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/courses/00000000-0000-0000-0000-000000000000/modules?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/courses/00000000-0000-0000-0000-000000000000/modules?
page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/courses/00000000-0000-0000-0000-
000000000000/modules?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/courses/00000000-0000-0000-0000-
000000000000/modules?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/courses/00000000-
0000-0000-000000000000/modules?page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
```

```
    if err != nil {  
        panic(err)  
    }  
    defer res.Body.Close()  
  
    if res.StatusCode >= 400 {  
        panic(fmt.Sprintf("API returned %d", res.StatusCode))  
    }  
  
    var body map[string]any  
    json.NewDecoder(res.Body).Decode(&body)  
    fmt.Println(body)  
}
```

GET /api/reporting/v1/students

List students with activity

Contacts with sign-in, session, and revenue metrics attached.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1–100 range.
role	query	string	Only students with this role.
sort	query	string	Pass 'last_active_at' to order by most recently active.

RESPONSE FIELDS — STUDENT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	The contact id. Same identifier the public API uses for contacts.
name	string null	Display name.
email	string	Email address.
role	string null	Your own classification of this contact.
joined_at	string null	When the contact was created. ISO-8601, UTC.
last_active_at	string null	Most recent recorded activity. ISO-8601, UTC.
logins_30d	integer	Sign-ins in the last 30 days.
avg_session_minutes	number	Mean session length in minutes over the last 30 days, to one decimal place.
total_session_minutes_30d	integer	Total session minutes over the last 30 days.
paying	boolean	True when this contact has ever bought.
lifetime_revenue_cents	integer	Total revenue attributed to this contact, in cents.

RESPONSES

- **200** — A page of students.
- **401** — The token is missing, malformed, invalid, or revoked.
- **429** — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/students?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/students?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})
```

```

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)

```

PYTHON

```

import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/students?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/students?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/students?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

```

```

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/students?
page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}

```

GET /api/reporting/v1/students/{student_id}

Get one student

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
student_id *	path	string (uuid)	The contact id.

RESPONSE FIELDS — STUDENT

FIELD	TYPE	DESCRIPTION
id	string (uuid)	The contact id. Same identifier the public API uses for contacts.
name	string null	Display name.
email	string	Email address.
role	string null	Your own classification of this contact.
joined_at	string null	When the contact was created. ISO-8601, UTC.
last_active_at	string null	Most recent recorded activity. ISO-8601, UTC.
logins_30d	integer	Sign-ins in the last 30 days.
avg_session_minutes	number	Mean session length in minutes over the last 30 days, to one decimal place.
total_session_minutes_30d	integer	Total session minutes over the last 30 days.
paying	boolean	True when this contact has ever bought.
lifetime_revenue_cents	integer	Total revenue attributed to this contact, in cents.

RESPONSES

- 200 — The student.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No student with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}
```

```
const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")
```

```
res, err := http.DefaultClient.Do(req)
if err != nil {
    panic(err)
}
defer res.Body.Close()

if res.StatusCode >= 400 {
    panic(fmt.Sprintf("API returned %d", res.StatusCode))
}

var body map[string]any
json.NewDecoder(res.Body).Decode(&body)
fmt.Println(body)
}
```

GET /api/reporting/v1/students/{student_id}/watch-history

List what a student has watched

Every lesson with recorded watch time for this student, newest first.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
student_id *	path	string (uuid)	The contact id.
page	query	integer	Page number, starting at 1. Values below 1 are clamped to 1.
page_size	query	integer	Rows per page. Clamped to the 1-100 range.

RESPONSE FIELDS — WATCHHISTORYENTRY

FIELD	TYPE	DESCRIPTION
course_id	string null	The course the lesson belongs to.
course_title	string null	Course title.
module_id	string null	The module the lesson belongs to.
module_title	string null	Module title.
lesson_id	string (uuid)	The lesson id.
lesson_title	string	Lesson title.
lesson_duration_seconds	integer null	Lesson length in seconds, when known.
watched_seconds	integer	Seconds of this lesson the student has watched.
watched_pct	number	Watched seconds as a percentage of duration, 0-100.
watched_over_30s	boolean	True when watch time exceeds 30 seconds.
last_watched_at	string null	Most recent watch event for this lesson. ISO-8601, UTC.

RESPONSES

- 200 — A page of watch history.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No student with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000/watch-history?page=1&page_size=50' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000/watch-history?page=1&page_size=50', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})
```

```

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)

```

PYTHON

```

import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000/watch-
history?page=1&page_size=50",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-
000000000000/watch-history?page=1&page_size=50');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-
000000000000/watch-history?page=1&page_size=50')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"

```

```

)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/students/00000000-0000-0000-0000-000000000000/watch-history?page=1&page_size=50", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}

```

GET /api/reporting/v1/lessons

Rank lessons by earliest drop-off

Lessons ordered by how early their audience falls away — earliest drop first, larger drops breaking ties. Lessons with no retention data are excluded. This endpoint takes `limit` rather than `page\_size`, and always reports page 1.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
sort	query	string	Only `worst_retention` is supported. Any other value returns 400.
limit	query	integer	How many lessons to return. Clamped to the 1-100 range.

RESPONSE FIELDS — WORSTRETENTIONLESSON

FIELD	TYPE	DESCRIPTION
lesson_id	string (uuid)	The lesson id.
title	string	Lesson title.
biggest_dropoff	Dropoff	

RESPONSES

- 200 — The ranked lessons.
- 400 — An unsupported `sort` value.
- 401 — The token is missing, malformed, invalid, or revoked.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/lessons?sort=worst_retention&limit=10' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/lessons?
sort=worst_retention&limit=10', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
```

```

    "https://app.zinstruct.com/api/reporting/v1/lessons?sort=worst_retention&limit=10",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())

```

PHP

```

<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/lessons?sort=worst_retention&limit=10');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/lessons?sort=worst_retention&limit=10')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/lessons?
sort=worst_retention&limit=10", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }
}

```

```
var body map[string]any
json.NewDecoder(res.Body).Decode(&body)
fmt.Println(body)
}
```

GET /api/reporting/v1/lessons/{lesson_id}/retention

Get a lesson's retention curve

The full drop-off curve for one lesson, plus the single biggest fall.

PARAMETERS

NAME	IN	TYPE	DESCRIPTION
lesson_id *	path	string (uuid)	The lesson id.

RESPONSE FIELDS — LESSONRETENTION

FIELD	TYPE	DESCRIPTION
lesson_id	string (uuid)	The lesson id.
title	string	Lesson title.
course_id	string null	The course the lesson belongs to.
module_id	string null	The module the lesson belongs to.
duration_seconds	integer null	Lesson length in seconds, when known.
views	integer	Recorded views.
avg_watch_seconds	number	Mean seconds watched per view.
avg_watch_pct	number	Mean percentage of the lesson watched, 0-100.
retention	RetentionPoint[]	The curve, ordered by offset.
biggest_dropoff	Dropoff	

RESPONSES

- 200 — The curve.
- 401 — The token is missing, malformed, invalid, or revoked.
- 404 — No lesson with that id.
- 429 — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/lessons/00000000-0000-0000-0000-000000000000/retention' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/lessons/00000000-0000-0000-0000-000000000000/retention', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}
```

```
const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/lessons/00000000-0000-0000-0000-000000000000/retention",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/lessons/00000000-0000-0000-0000-000000000000/retention');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/lessons/00000000-0000-0000-0000-000000000000/retention')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/lessons/00000000-0000-0000-0000-000000000000/retention", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")
```

```
res, err := http.DefaultClient.Do(req)
if err != nil {
    panic(err)
}
defer res.Body.Close()

if res.StatusCode >= 400 {
    panic(fmt.Sprintf("API returned %d", res.StatusCode))
}

var body map[string]any
json.NewDecoder(res.Body).Decode(&body)
fmt.Println(body)
}
```

GET /api/reporting/v1/summary

Get the KPI summary

A single flat object of headline numbers for your school — the cheapest call for a dashboard tile.

RESPONSE FIELDS — SUMMARY

FIELD	TYPE	DESCRIPTION
tenant	string	Your school slug.
students	integer	Total contacts.
active_students_30d	integer	Distinct contacts with activity in the last 30 days.
paying_customers	integer	Contacts who have bought.
courses	integer	Course products.
modules	integer	Modules across all courses.
lessons	integer	Lessons across all modules.
logins_30d	integer	Sign-ins in the last 30 days.
avg_session_minutes	number	Mean session length in minutes.
lessons_watched_over_30s_30d	integer	Lesson views over 30 seconds in the last 30 days.
completion_rate_pct	number	Completed enrollments as a percentage of all enrollments, 0–100.

RESPONSES

- 200 — The summary.
- 401 — The token is missing, malformed, invalid, or revoked.
- 429 — Rate limit exceeded for this key.
- 500 — The summary could not be built.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/summary' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/summary', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/summary",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/summary');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));
```

RUBY

```
require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/summary')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)
```

GO

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/summary", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()
```

```
if res.StatusCode >= 400 {  
    panic(fmt.Sprintf("API returned %d", res.StatusCode))  
}  
  
var body map[string]any  
json.NewDecoder(res.Body).Decode(&body)  
fmt.Println(body)  
}
```

Getting started

GET /api/reporting/v1/health

Check a key works

Returns `{ "ok": true }` when the key authenticated. The right first call when wiring up an integration: it proves the key and the header are correct without touching any data.

RESPONSE FIELDS — HEALTH

FIELD	TYPE	DESCRIPTION
ok	boolean	Always true when the key authenticated.

RESPONSES

- **200** — The key is valid.
- **401** — The token is missing, malformed, invalid, or revoked.
- **429** — Rate limit exceeded for this key.

CURL

```
curl -X GET 'https://app.zinstruct.com/api/reporting/v1/health' \
-H 'Authorization: Bearer YOUR_API_KEY'
```

JAVASCRIPT

```
const res = await fetch('https://app.zinstruct.com/api/reporting/v1/health', {
  method: 'GET',
  headers: { Authorization: 'Bearer YOUR_API_KEY' },
})

if (!res.ok) {
  const { error } = await res.json()
  throw new Error(`${error.code}: ${error.message}`)
}

const body = await res.json()
console.log(body)
```

PYTHON

```
import requests

res = requests.get(
    "https://app.zinstruct.com/api/reporting/v1/health",
    headers={"Authorization": "Bearer YOUR_API_KEY"},
    timeout=30,
)
res.raise_for_status()
print(res.json())
```

PHP

```
<?php
$ch = curl_init('https://app.zinstruct.com/api/reporting/v1/health');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_HTTPHEADER, ['Authorization: Bearer YOUR_API_KEY']);
```

```

$body = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

if ($status >= 400) {
    throw new RuntimeException("API returned $status: $body");
}

print_r(json_decode($body, true));

```

RUBY

```

require 'net/http'
require 'json'

uri = URI('https://app.zinstruct.com/api/reporting/v1/health')
req = Net::HTTP::Get.new(uri)
req['Authorization'] = 'Bearer YOUR_API_KEY'

res = Net::HTTP.start(uri.hostname, uri.port, use_ssl: true) { |http| http.request(req) }
raise "API returned #{res.code}: #{res.body}" unless res.is_a?(Net::HTTPSuccess)

puts JSON.parse(res.body)

```

GO

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"
)

func main() {
    req, _ := http.NewRequest("GET", "https://app.zinstruct.com/api/reporting/v1/health", nil)
    req.Header.Set("Authorization", "Bearer YOUR_API_KEY")

    res, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer res.Body.Close()

    if res.StatusCode >= 400 {
        panic(fmt.Sprintf("API returned %d", res.StatusCode))
    }

    var body map[string]any
    json.NewDecoder(res.Body).Decode(&body)
    fmt.Println(body)
}

```