Hands on Introduction to Gravitee.io API Management

Starting soon!

Have you got Gravitee APIM up and running yet?

- No? Instructions available here: dev.gravitee.io/local
- Follow ‘Setting up APIM’ instructions

Hands on Introduction to Gravitee.io API Management

Part one

- What is a “Design First” approach?
  - What are the benefits of Design First?
- Introducing the bookshop
  - Who are the users and what are the requirements?

Part two

- Overview of API Management
- Overview of Gravitee API Management platform
- Creating and publishing the API based on the bookshop
- Creating documentation
- Adding policies and plans
- Where to learn more

Housekeeping

Questions

- Ask them in the chat, we'll try to answer them as we go along

Is this session recorded?

- Yes it is! You'll be able to re-watch the session after its conclusion

Session study group?

- We'll create one under 'Conversation Corner' on the community forum: **community.gravitee.io**

Before we get started!

You should have already installed and got Gravitee APIM up and running. If you haven't, there is still time

- Follow the instructions in the following blog post under 'Setting up APIM'
  - **dev.gravitee.io/local**

Design First

What is API First?

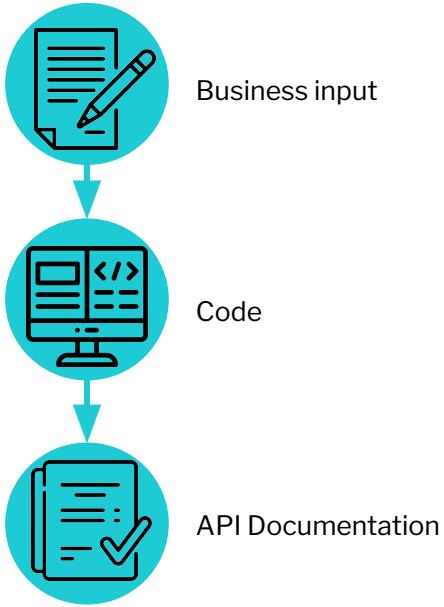
- Treat APIs as first class citizens
- Consideration that everything in the project will ultimately use APIs
- Develop APIs ahead of other components

Advantages

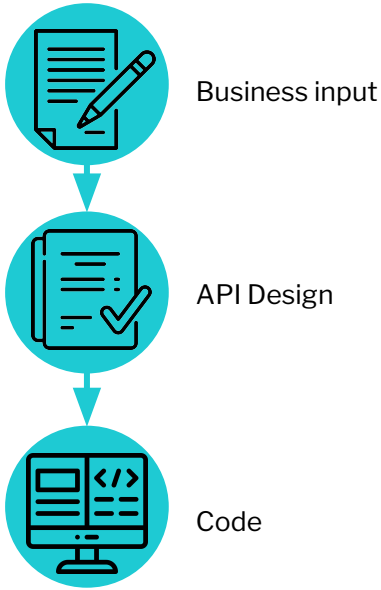
- Reusability
- Stakeholder collaboration
- Lower app development costs

API First is not always Design First

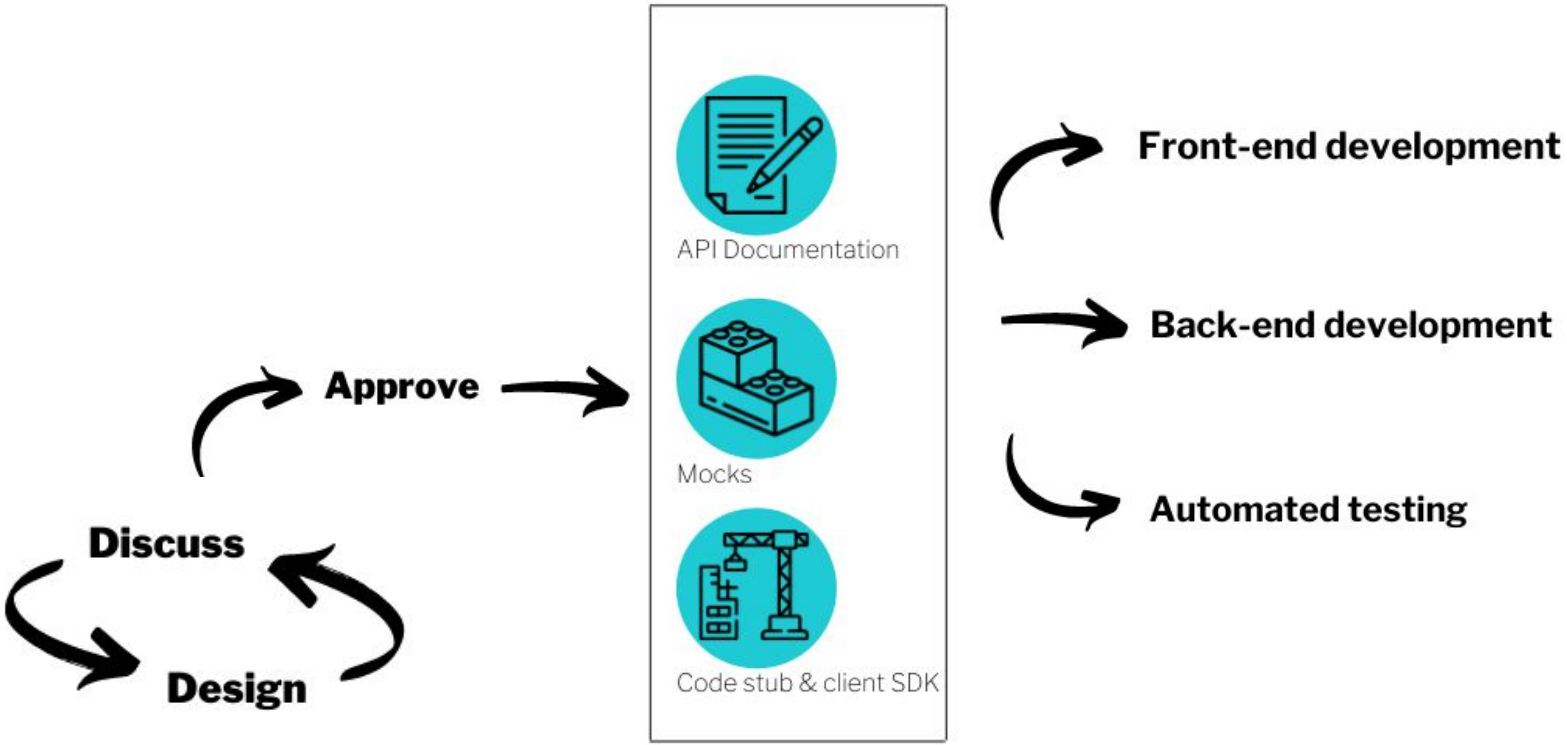
Code First



Design First



Design First API lifecycle



Icons by Freepik https://www.freepik.com

Benefits of using Design First approach


Improved communication


Rapid feedback loop from business & customers

Product-driven APIs


Better developer experience


Help drive parallel development


Enforced governance and security

The simple bookshop (Design First!)

Introducing “Simple books”

- Online only bookshop
- Sells many book titles
- Easy access browsing facility
- Aims to ship orders the following day

Book shop requirements (cut down!)

Who are the users?

- **Browsers** - anonymous visitors viewing inventory
- **Storeroom Dispatcher** - employee that posts the orders

(For the purposes of this exercise, we're only picking two specific users + scenario!)

Renee Fisher on Unsplash

Book shop requirements (cut down!)

What are the requirements?

- Show all titles currently available in inventory
- Show what titles need to be shipped to a specific address to logged in employees

Non-functional requirements

- Rate limit how often anonymous users can view inventory
- Set a specific location from where employees can log in to the order system

Book shop requirements (cut down!)

API model

- Books
  - View available books (GET) /books
  - View information about a particular book
- Orders
  - View titles, order numbers, name and shipping address (GET) /orders
  - View information about a particular order

Renee Fisher on Unsplash

Brief OpenAPI overview

- Open-source format for describing and documenting RESTful APIs
- OpenAPI allows us to provide:
  - descriptive information (meta),
  - end points (path items)
  - define reusable components (e.g. schemas, parameters, examples, etc.)
- Latest version (3.0) can be written in JSON or YAML

https://www.openapis.org/

Renee Fisher on Unsplash

Our API schema

- Version 1.0.0
- Root /bookshop-<number>
- /bookshop/books
    - GET method, searchable on id
    - Attributes: id:integer, title:string, author:string
    - Examples: id:1, title: “A tale of two cities”, author: “Charles Dickens”
- /bookshop/orders
    - GET method, searchable on id
    - Attributes: id:integer, book-id:integer, name:string, address:string, date:string
    - Examples: id:100, book-id:1, name:”Jo Bloggs”, address:”12 North Street”, date:”2022-01-01”

API Management

What is API Management?

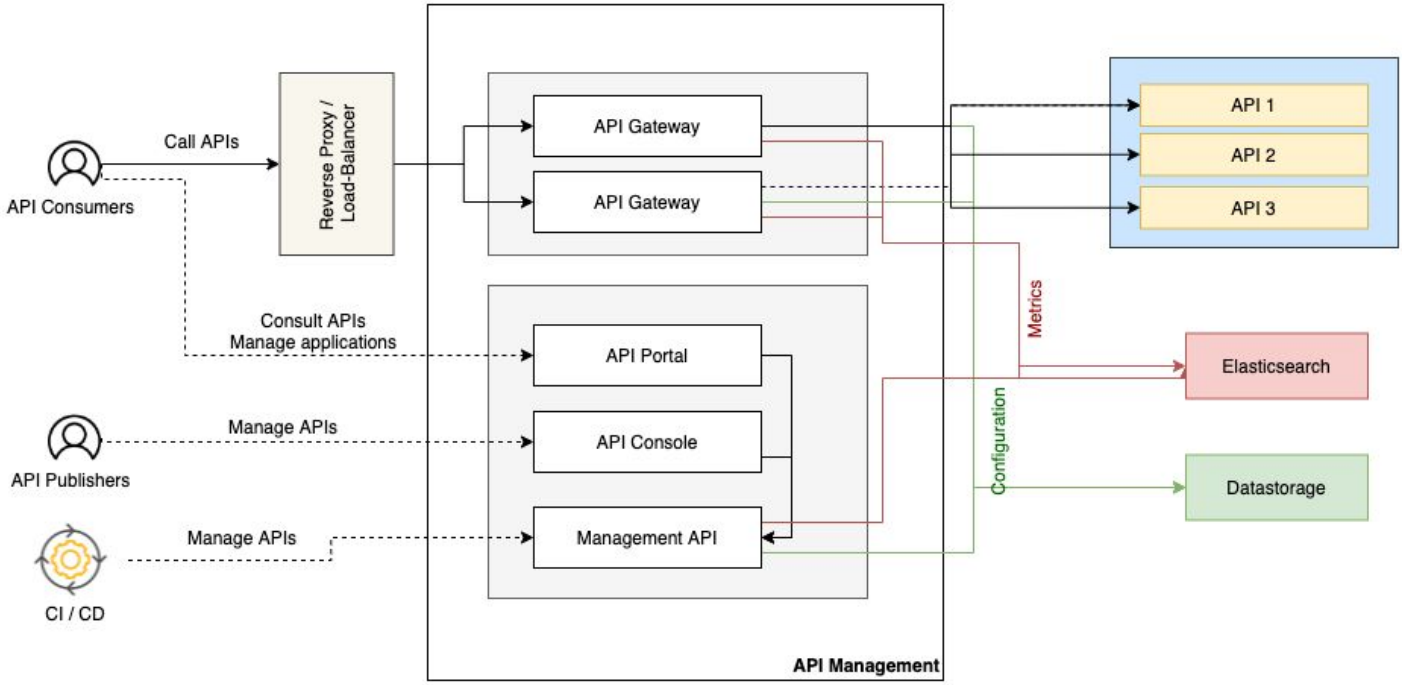
API Management is

- process of creating and publishing APIs
- enforcing their usage policies
- controlling access
- nurturing the subscriber community
- collecting and analyzing usage statistics

Why use API Management?

- Dynamic and rapid changes based on new requirements
- ‘One stop shop’ for authentication, authorisation and access control
- Easier to manage the lifecycle of an API

Gravitee APIM architecture and standard deployment



API lifecycle in Gravitee.io APIM

- **Develop** - Create and Publish an API
- **Documentation** - Developers can see the documentation of your APIs
- **Secure** - Control how your APIs are accessed and consumed through plans and policies
- **Monitor** - Monitor your API usage from your consumers

To consume an API, developers must:

- Create an application linked to an API plan (unless keyless)
- Subscribe to the API
    - Based on workflow, APIM will accept/deny the request

You turn - importing the API schema

Have a go at importing the API schema, making sure that the mocks and the documentation are generated, then try to deploy and call the API endpoints

Steps:

- Download the API schema from **dev.gravitee.io/bookshop**
- Go to APIs (console address: localhost:8084 username/password admin/admin)
- Create a new API → Import
- Make sure you select Mocks, Docs, etc.
- Create a keyless plan and publish (name browse)
- Deploy everything and try calling the endpoints

Don’t panic if you miss something, we’ll show you how to sort and missed steps

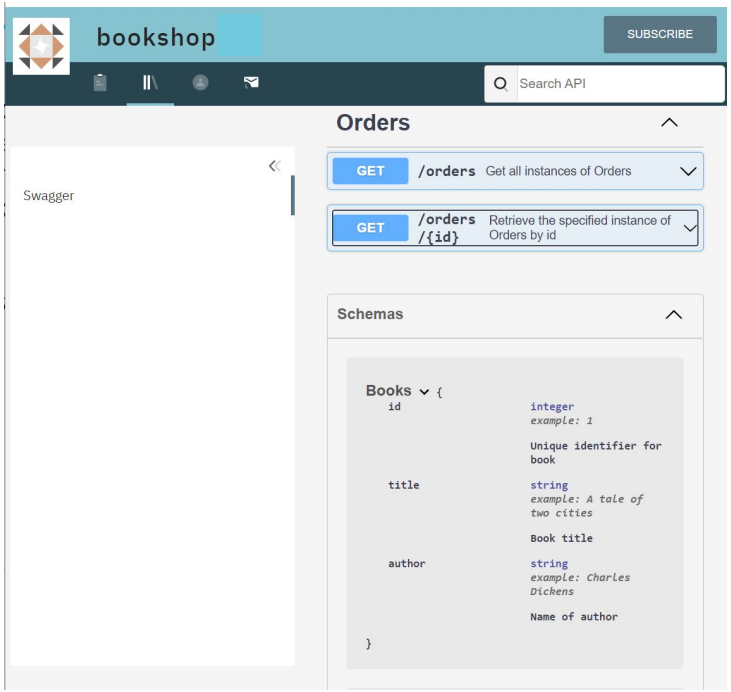
(gateway address: localhost:8082 - e.g. localhost:8082/bookshop/books)

Documentation

Customising documentation

- Documentation generated by API schema
  - Either by importing and selecting the options
  - Or by going to docs and generate/publish
- Also possible to try out examples
  - Enable ‘Try’ in docs

Can also create custom pages using MarkDown and AsciiDoc, as well as importing/editing Swagger and AsyncAPI specifications



You turn - create a documentation home page

- Create a new home page
- Pick your favourite markup language (e.g. MarkDown, AsciiDoc, etc.)
- Create a simple page, and then publish and view it
- If you have not already done so, you will need to publish your API to view the API and associated documentation

Example content for the page:

- Name of the bookshop
- Brief description of what the bookshop does
- Brief description of what services are available
- An image (unsplash is a good source)

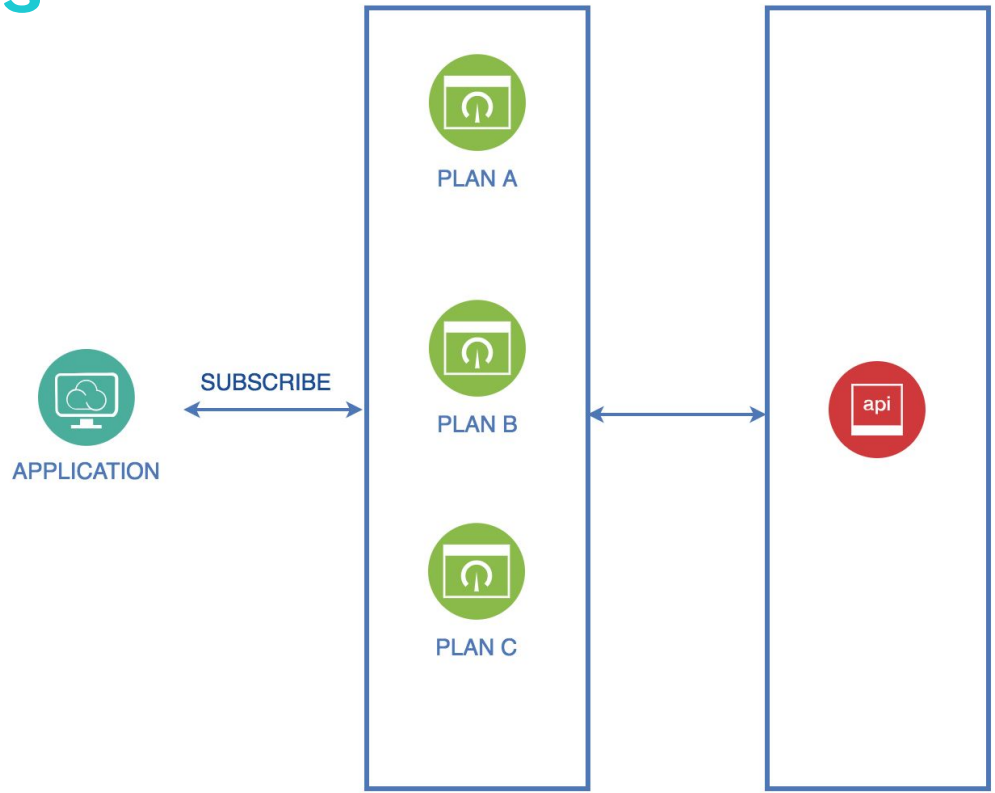
View the pages on the Developer Portal - localhost:8085

Plans and Applications

Plans and subscriptions

Once an API is registered and made public, you can manage subscriptions to it through APIM plans. Managing subscriptions and plans is a key feature of APIM that publishers can use to provide and regulate access to APIs

A plan provides a service and access layer on top of your APIs for consumer applications. A plan specifies access limits, subscription validation modes and other configuration to tailor it to a specific application.



What is a Gravitee API Plan ?

- API as a Product
  - Apply read-only access and limit request traffic as part of API discovery
  - Differentiate API experience based on user groups
  - Apply business rules by default across all endpoints, rather than policies on each one
- The contract between the Application and the API
- Adds the security
- Allows you add broad-brush policies such as rate limiting and IP filtering
- To consume an API, you need a plan

Your turn - Create a Plan for the employees

In the APIM Console, create the following plan :

- name: **Employee login**
- Description: **Employee only**
- **NEXT**
- Add API Key Authentication
- **NEXT**
- **Save**
- **Publish the Plan / Deploy the API**
- **Test in your favourite REST tool**

One more thing...

It doesn't work in the gateway

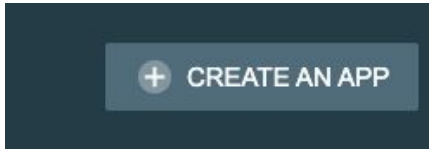
- We need to make a change to the plans
- At the moment everything would go to anonymous plan
- Add a filter policy to the anonymous plan
    - In the blacklist, add the endpoint for /orders

What are applications?

- A way for consumers to use an API
- A way for API publishers to control and regulate access to their APIs
- Typical applications are web apps, native applications, and bash/jobs which access data
- To consume an API with a non-keyless plan, you need an application

Applications can be managed from both APIM Console and Portal

Your turn - create an applications



Name of Application : app_bokshop

Description : Test description

Type : Web

Find your API : Search for your bookshop + number

Subscribe

Go to Subscriptions

Accept the request and copy the API Key

Go to Your REST query tool of choice

Add a header **X-Gravitee-Api-Key** and paste the API Key

Now the API will return echo data

cURL option

curl --header "X-Gravitee-Api-Key: <API key>" http://localhost:8082/bookshop/orders

API Flows and Policies

What are policies and API Flows?

Policies:

- Modifies the behaviour of the request or response flow
- Allows you to apply inbound/outbound rules
- “Proxy controller” - guaranteeing a business rule is fulfilled
- Allows you to apply inbound/outbound rules
- Can be chained using a logical order
- Can be configured on a plan and/or a couple HTTP path/verb
- Supports **Expression Language (EL)**, including properties (manual & dynamic)
- Uses resources

API Flows:

- Used for plans and to define policies for each flow
- Creating different flows for a plan allows you to apply different policies by path and/or HTTP method

Types of policies

Security

- Api-key, OAuth2, JWT / JWS, IP filtering, Resource filtering, CORS, Rate Limit / Quota, Content Limit, Request validation

Performance

- Cache

Transformation

- Headers, Query Params, Rest / Soap, HTML / JSON, XML / JSON, JSON / JSON, XSLT, Override HTTP method

Other

- Mocks and dynamic routing

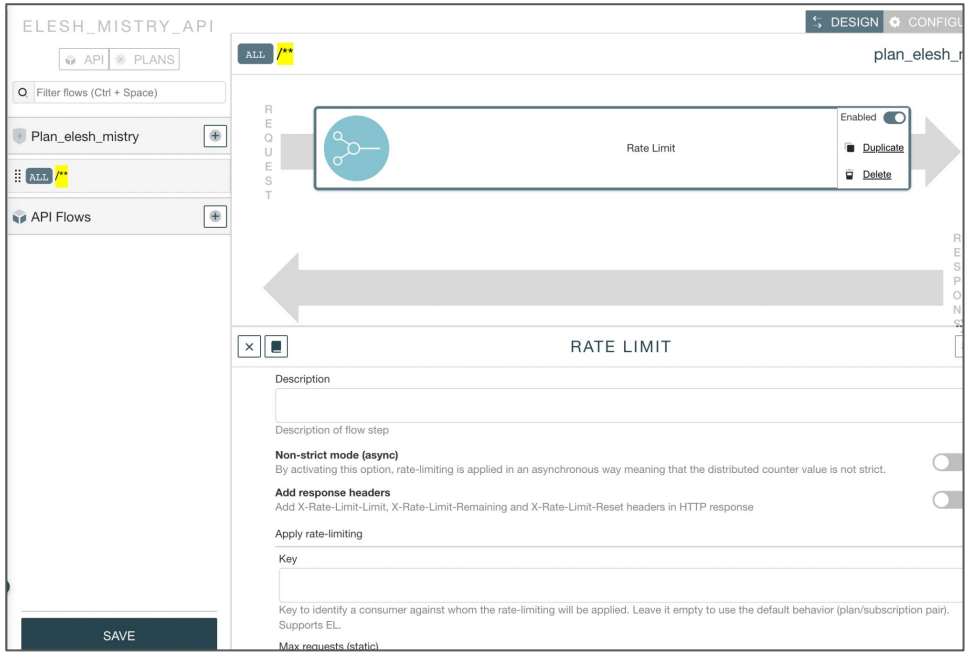
Your turn - create a new mock policy

- Add the mock policy on a new path
  - Create a path named: **/mock**
  - Drag and drop the mock policy
  - Configure the mock
- Make a request on the endpoint /mock:

localhost:8082/bookshop/mock

Your turn - add a rate limit policy

- Why would we want to rate limit requests?
- Add a rate limiting policy to the keyless plan and test it out



Your turn - add IP filtering for orders

- Why would we want to route requests based on location?
- Configure to make the plan for employees only accessible in the UK (192.168.0.0/16)
- (Or try configuring to your IP range!)
- We can test it's working by trying a different IP range - take your pick!

What next?

What next?

- Try another specification
- We'd love to hear your feedback!
  - **dev.gravitee.io/march-training**
- YouTube videos
  - **dev.gravitee.io/video**
- Blogs
  - **gravitee.io/blog**
- Join the community forum
  - **community.gravitee.io**
  - We'll keep you updated with new content, videos, events and training!
  - Ask your questions!