

Latest API Mapping with endpoint

APIs API	
Endpoint	Description
GET/apis	List in-house APIs
POST/apis	Add a new API
DELETE/apis/{apild}	Remove a legacy API
PUT/apis/{apild, name }	Modify a current API

Service API	
Endpoint	Description
GET/services	List all services
POST/services	Add a new service
PUT/services/{serviced, name }	Update the service
DELETE/services/{serviceld}	Delete the service

User API	
Endpoint	Description
GET/users/valid/{username}	Validate username
GET/users/valid/{password}	Validate password
POST/users/{ name, email, username, password }	Create a new profile
GET/users/{userId}	Find user profile
DELETE/users/{userId}	Delete user profile

System API	
Endpoint	Description
GET/systems	Fetch system version
POST/systems/latest	Install the latest release

Event API	
Endpoint	Description
GET/events	List all events
POST/events/{ name, datetime }	Add a new event
PUT/events/{eventId, name }	Update the event
DELETE/events/{eventId}	Delete the event
GET/events/{datetime}	Get events on a given date

Datetime API	
--------------	--

Endpoint	Description
GET/datetime/date	Get the current date
GET/datetime/day	Get the current day
GET/datetime/month	Get the current month
GET/datetime/year	Get the current year

1. What were some of the alternative design options considered? Why did you choose the selected option?

Let me discuss alternative design options considered both in terms of RESTful APIs framework and endpoint testing tool. I chose to use Spring Boot to develop my MVC framework because it is one of the mainstream frameworks that make it easy to create stand-alone, production-grade applications. Also, it would be a great opportunity for me to explore as I never use it before. As for rest-assured, it provides simple APIs that allows me to test the functionality of each endpoint's status code as well as content directly. I am sure there are plenty of other frameworks and endpoint testing tools exist such as Restlet, SparkJava, but Spring Boot and rest-assured are what I recommend for this assignment

2. What changes did you need to make to your tests (if any) to get them to pass. Why were those changes needed, and do they shed any light on your design?

A1.2 is based on A1.1 with **24** endpoints in total, but some of endpoints may require passing in additional parameters, highlighted in **yellow** above. These changes are needed to incorporate lack of vision in early stage of design. Therefore, we can conclude that design and implementation are highly dependent. Specifically, the initial design will shape implementation and implementation can validate design.

3. Pick one design principle discussed in class and describe how your design adheres to this principle.

One of design principle I found particularly important is Test Driven Development (TDD), which means we wrote some test cases prior to the design. These become the criteria for determining if the design is suitable. And the process of implementing API will help validate API. For example, we may miss some parameters that are required passing in at the design phrase. In addition, we can keep the code as test cases and add more test cases if needed.