**Designing REST APIs: Part 1**

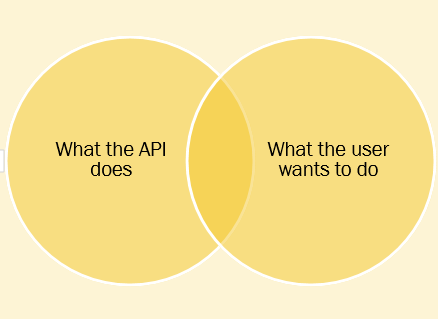
*Posted On August 28, 2019*

This post will help users to design an API while considering HTTP concepts and REST constraints.

**The Key Aspect of API Design**

* Decide what to expose
* Decide how to expose
* Test and validate assumptions
* Iterate

There has to be a confluece of what API does and what the user wants to do.



**Approach to add an API**

1. Bolt-On Strategy: The implementation comes to use when you have an existing application and add an API after that. It is considered as Brute-force approach, being the fastest way to get value from an API as code and system are existing whole time.
2. Greenfield Strategy: It is a new system. Users have complete freedom and fexibility. It is easy way to develop an API. It takes advantage of new technologies and architectures.
3. Facade Strategy: This approach for building an API comes between Bolt-On and Greenfield Strategy. It takes advantage of existing system and shape it on your requirements.

**API Modeling** **Proces**s



*Identify Participants*: Find out the people and entity who will use API. The people included could be internal or external, active or passive.

*Identify Activities*: These are story points that are captured while set of event take place. These are recorded as steps.

*Activity Steps*: Once participants and activities are recorded, break them down into steps.

*Create API Definitions*: Identify the resources, get HTTP verbs based on Item Resource.

Suppose in a shopping cart system of a website, you include features containing nouns as identified here:

* View Items
* Add items to cart
* Add other items/ cancel item
* Check out
* Fulfill and ship order
* View Order

List of action i.e. Resources are:-

* List items
* View item
* Add items to cart
* View cart
* Check out
* List orders
* View orders
* Cancel orders

We have to map nouns to http verbs. Based on CRUD operations of Create, Read, Update, Delete and the basic verbs for these as Get, Put, Delete and Post.

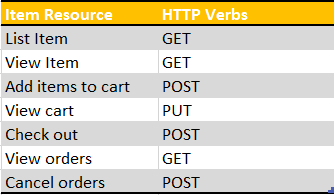
**HTTP Verbs**:

GET: Used for retieval of resource or list of resources.

DELETE: Used to delete a resource

PUT: Used to update an existing resource

POST: Create a new resource and change the state of resource.



*Validate API*: Accept incoming requests, Validate its verbs and URL patterns, Return static HTTP response codes.