**62.1. Spring REST - API Design Best Practices**

**REST API Design**:

For REST API design, when we will be building a real-time project, we will need to know

* For real-time projects, who will use your API?
* Also, how will they use your API?
* Design the API based on requirements

**API Design Process (Step-by-Step)**:

1. Review API requirements
2. Identify main resource / entity
3. Use HTTP methods to assign action on resource

**Step 1: Review API Requirements**:

Create a REST API for the Customer Relationship Management (CRM) system. REST clients should be able to

* Get a list of customers
* Get a single customer by id
* Add a new customer
* Update a customer
* Delete a customer

**Step 2: Identify main resource / entity**:

* To identify main resource / entity, look for the most prominent "noun"
* For our project, it is "customer"
* Convention is to use plural form of resource / entity: **customers**

api/customers

**Step3: Use HTTP methods to assign action on resource**:

|  |  |
| --- | --- |
| HTTP Method | CRUD Action |
| POST | Create a new entity |
| GET | Read a list of entities or single entity |
| PUT | Update an existing entity |
| DELETE | Delete an existing entity |

**CRUD Endpoint Examples**:

|  |  |  |
| --- | --- | --- |
| HTTP Method | Endpoints | CRUD Action |
| POST | /api/customers | Create a new customer |
| GET | /api/customers | Read a list of customers |
| GET | /api/customers/{customerId} | Read a single customer |
| PUT | /api/ customers | Update an existing customer |
| DELETE | /api/customers/{customerId} | Delete an existing customer |

**Note**:

For **POST** and **PUT**,we will send customer data as JSON in request message body.

**Anti-Patterns**:

Don’t do the following, this are REST anti-patterns, bad practice.

/api/customersList

/api/deleteCustomer

/api/addCustomer

/api/updateCustomer

Don’t include the action in the endpoint. Instead, use HTTP methods to assign actions. Endpoints only entity/resource name. No actions.

**More API example**:

* PayPal Invoicing API

<https://developer.paypal.com/docs/api/invoicing/>

* GitHub

User: <https://api.github.com/user>

Repo: [https://api.github.com/users/{username}/repos](https://api.github.com/users/%7busername%7d/repos)

**Note**:

In REST API notation (:) for path variables is same as {}. "/repos/:owner/:repo" and "/repos/{owner}/{repo}" is same.

62.1. Spring REST - API Design Best Practices