

# Spring REST API Design



# REST API Design

# REST API Design

- For real-time projects, who will use your API?

# REST API Design

- For real-time projects, who will use your API?
- Also, how will they use your API?

# REST API Design

- 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*

# API Design Process

*Step-By-Step*

## 1. Review API requirements

# API Design Process

*Step-By-Step*

1. Review API requirements
2. Identify main resource / entity

# 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

# Step 1: Review API Requirements

From the Boss

Create a REST API for the  
Customer Relationship Management (CRM) system

# Step 1: Review API Requirements

From the Boss

Create a REST API for the Customer Relationship Management (CRM) system

REST clients should be able to

# Step 1: Review API Requirements

From the Boss

Create a REST API for the Customer Relationship Management (CRM) system

REST clients should be able to

- Get a list of customers

# Step 1: Review API Requirements

From the Boss

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

# Step 1: Review API Requirements

From the Boss

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

# Step 1: Review API Requirements

From the Boss

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

# Step 1: Review API Requirements

From the Boss

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 1: Review API Requirements

From the Boss

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

Full  
CRUD

# Step 2: Identify main resource / entity

# Step 2: Identify main resource / entity

- To identify main resource / entity, look for the most prominent "noun"

# Step 2: Identify main resource / entity

- To identify main resource / entity, look for the most prominent "noun"
- For our project, it is "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**

# 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
```

# Step 3: Use HTTP methods to assign action on resource

# Step 3: Use HTTP methods to assign action on resource

**HTTP Method**

**CRUD Action**

# Step 3: Use HTTP methods to assign action on resource

HTTP Method	CRUD Action
POST	<u>C</u> reate a new entity

# Step 3: Use HTTP methods to assign action on resource

HTTP Method	CRUD Action
POST	<u>C</u> reate a new entity
GET	<u>R</u> ead a list of entities or single entity

# Step 3: Use HTTP methods to assign action on resource

HTTP Method	CRUD Action
POST	<u>Create a new entity</u>
GET	<u>Read a list of entities or single entity</u>
PUT	<u>Update an existing entity</u>

# Step 3: Use HTTP methods to assign action on resource

HTTP Method	CRUD Action
POST	<u>Create a new entity</u>
GET	<u>Read a list of entities or single entity</u>
PUT	<u>Update an existing entity</u>
DELETE	<u>Delete an existing entity</u>

# Step 3: Use HTTP methods to assign action on resource

HTTP Method	CRUD Action
POST	<u>Create</u> a new entity
GET	<u>Read</u> a list of entities or single entity
PUT	<u>Update</u> an existing entity
DELETE	<u>Delete</u> an existing entity

Full  
CRUD

# CRUD Endpoint Examples

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
-------------	----------	-------------

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	Create a new customer

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>
GET	/api/customers/{customerId}	<u>Read a single customer</u>

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>
GET	/api/customers/{customerId}	<u>Read a single customer</u>
PUT	/api/customers	<u>Update an existing customer</u>

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>
GET	/api/customers/{customerId}	<u>Read a single customer</u>
PUT	/api/customers	<u>Update an existing customer</u>
DELETE	/api/customers/{customerId}	<u>Delete an existing customer</u>

# CRUD Endpoint Examples

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>
GET	/api/customers/{customerId}	<u>Read a single customer</u>
PUT	/api/customers	<u>Update an existing customer</u>
DELETE	/api/customers/{customerId}	<u>Delete an existing customer</u>

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

# Anti-Patterns

# Anti-Patterns

- DO NOT DO THIS ... these are REST anti-patterns, bad practice

# Anti-Patterns

- DO NOT DO THIS ... these are REST anti-patterns, bad practice



**/api/customersList**

**/api/deleteCustomer**

**/api/addCustomer**

**/api/updateCustomer**

# Anti-Patterns

- DO NOT DO THIS ... these are REST anti-patterns, bad practice



/api/customersList  
/api/deleteCustomer  
/api/addCustomer  
/api/updateCustomer



Don't include actions in the endpoint

# Anti-Patterns

- DO NOT DO THIS ... these are REST anti-patterns, bad practice

/api/customersList  
/api/deleteCustomer  
/api/addCustomer  
/api/updateCustomer



Don't include actions in the endpoint

Instead, use  
HTTP methods  
to assign actions



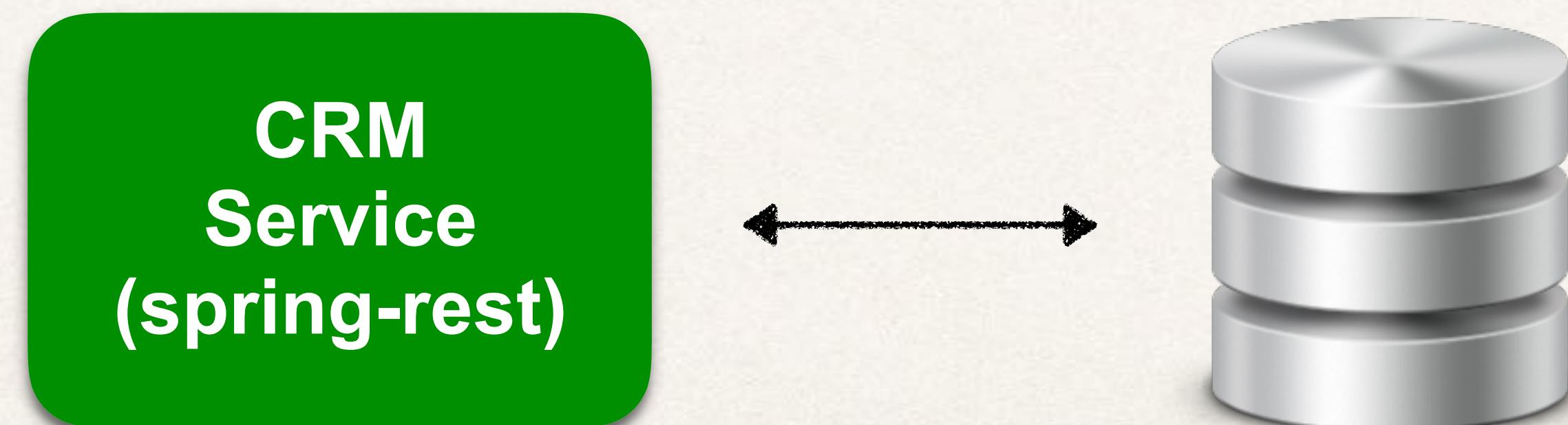
# CRM Real-Time Project

# CRM Real-Time Project

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>
GET	/api/customers/{customerId}	<u>Read a single customer</u>
PUT	/api/customers	<u>Update an existing customer</u>
DELETE	/api/customers/{customerId}	<u>Delete an existing customer</u>

# CRM Real-Time Project

HTTP Method	Endpoint	CRUD Action
POST	/api/customers	<u>Create a new customer</u>
GET	/api/customers	<u>Read a list of customers</u>
GET	/api/customers/{customerId}	<u>Read a single customer</u>
PUT	/api/customers	<u>Update an existing customer</u>
DELETE	/api/customers/{customerId}	<u>Delete an existing customer</u>



# More API Examples

# More API Examples

- On the following slides, we'll look at APIs from other real-time projects

# More API Examples

- On the following slides, we'll look at APIs from other real-time projects
- PayPal

# More API Examples

- On the following slides, we'll look at APIs from other real-time projects
- PayPal
- GitHub

# More API Examples

- On the following slides, we'll look at APIs from other real-time projects
- PayPal
- GitHub
- SalesForce

# PayPal



# PayPal



- PayPal Invoicing API

# PayPal



- PayPal Invoicing API
  - <https://developer.paypal.com/docs/api/invoicing/>

# PayPal

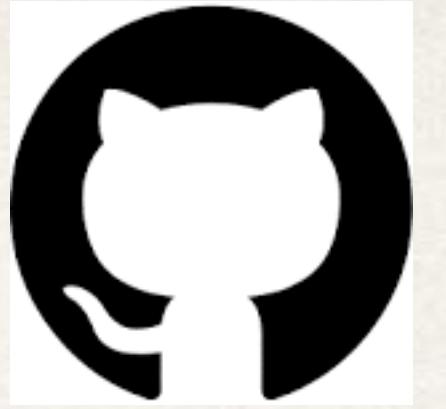


- PayPal Invoicing API
  - <https://developer.paypal.com/docs/api/invoicing/>

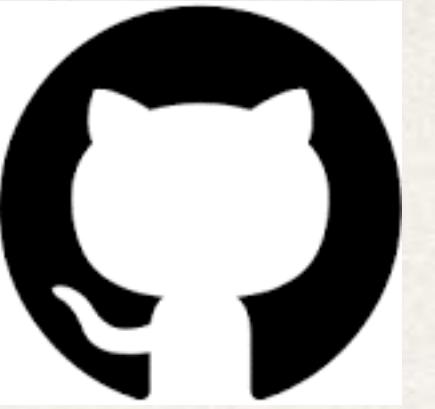
A screenshot of the PayPal Developer API documentation for Invoicing. The top navigation bar includes the PayPal logo, 'Developer', 'Docs', 'APIs', and 'Support'. Below the navigation, there are five main sections: 'Create draft invoice' (POST /v1/invoicing/invoices), 'Update invoice' (PUT /v1/invoicing/invoices/{invoice\_id}), 'List invoices' (GET /v1/invoicing/invoices), 'Delete draft invoice' (DELETE /v1/invoicing/invoices/{invoice\_id}), and 'Show invoice details' (GET /v1/invoicing/invoices/{invoice\_id}). Each section contains a colored button indicating the HTTP method and the API endpoint path.

- Create draft invoice**  
**POST** /v1/invoicing/invoices
- Update invoice**  
**PUT** /v1/invoicing/invoices/{invoice\_id}
- List invoices**  
**GET** /v1/invoicing/invoices
- Delete draft invoice**  
**DELETE** /v1/invoicing/invoices/{invoice\_id}
- Show invoice details**  
**GET** /v1/invoicing/invoices/{invoice\_id}

# GitHub

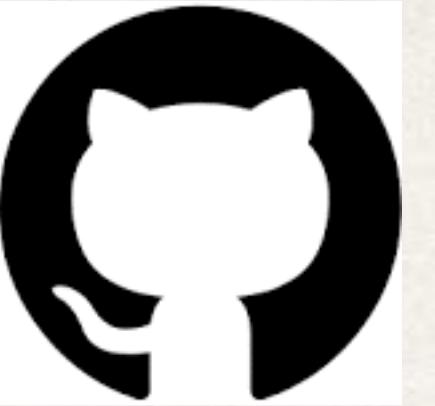


# GitHub



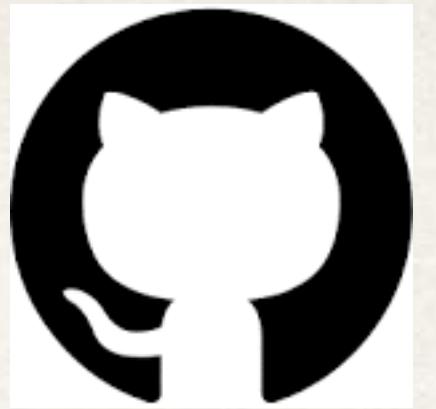
- GitHub Repositories API

# GitHub

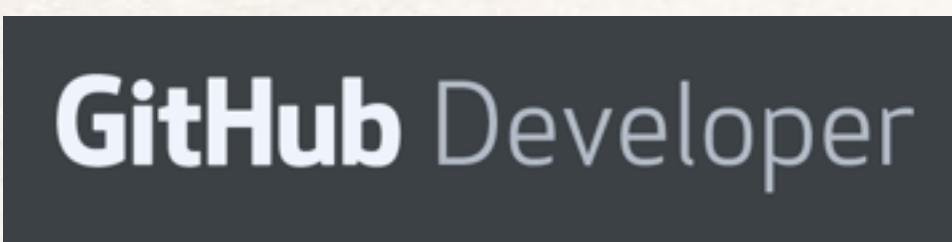


- GitHub Repositories API
  - <https://developer.github.com/v3/repos/#repositories>

# GitHub



- GitHub Repositories API
  - <https://developer.github.com/v3/repos/#repositories>



## Create a new repository

POST /user/repos

## Delete a repository

DELETE /repos/:owner/:repo

## List your repositories

GET /user/repos

## Get a repository

GET /repos/:owner/:repo

# SalesForce REST API



# SalesForce REST API

- Industries REST API



# SalesForce REST API

- Industries REST API
  - <https://sforce.co/2J40ALH>



# SalesForce REST API

- Industries REST API

- <https://sforce.co/2J40ALH>



## Retrieve All Individuals

**GET** /services/apexrest/v1/individual/

## Retrieve One Individual

**GET** /services/apexrest/v1/individual/{individual\_id}

## Create an individual

**POST** /services/apexrest/clinic01/v1/individual/

## Update an individual

**PUT** /services/apexrest/clinic01/v1/individual/