Rest API

miguel.garrido@ironhack.com

IRON
HACK

Creating rest APIs

Use **lowercase** letters when naming endpoints

Do not use underscores (_), use hyphens (-)

Here is an example of the commonly used URI structure that follows this naming rule:

```
1    <protocol>://<domain>/<context>/<resource-collection>/<resource-id>
```

Copy

✦ **Explain this code**

IRON
HACK

Creating rest APIs

Example | RESTful Endpoints

Following the above best practices, REST APIs should have endpoints structured along these lines:

HTTP verb	URL	Request body	Action
POST	`/api/orders`	JSON	Creates a new order
GET	`/api/orders`	(empty)	Returns all orders
PUT	`/api/orders/:id`	JSON	Edits a single order by id
DELETE	`/api/order/:id`	(empty)	Deletes a single order by id

IRON
HACK

Creating rest APIs

Use **nouns** instead of verbs in endpoint paths

Example

BAD ❌ - Below endpoint contains a verb: `get-order`.

```
1    GET    http://example.com/shop/get-order/3
```
✨ **Explain this code**

GOOD ✅ - Endpoint name contains only nouns, and the HTTP verb **GET** is used to specify the action.

```
1    GET    http://example.com/shop/orders/3
```
✨ **Explain this code**

IRON HACK

Creating rest APIs

Next Steps:

1. Make/Create the model

2. Create the Route

3. Test the Route on Postman

4. Enable CORS if needed

IRON
HACK

Bonus: Rest Routes with React

Action	Backend API Route	React Router URL (Frontend)	Axios Call (Frontend)
List all	`GET /api/projects`	`/projects`	`axios.get('/api/projects')`
Create	`POST /api/projects`	`/projects/new`	`axios.post('/api/projects', newProject)`
View one	`GET /api/projects/:id`	`/projects/:id`	`axios.get( /api/projects/${id} )`
Edit form	*Not in API*	`/projects/:id/edit`	`axios.get( /api/projects/${id} )`
Update	`PUT /api/projects/:id`	- (redirect after submit)	`axios.put( /api/projects/${id} , data)`
Delete	`DELETE /api/projects/:id`	- (redirect after submit)	`axios.delete( /api/projects/${id} )`

IRON
HACK