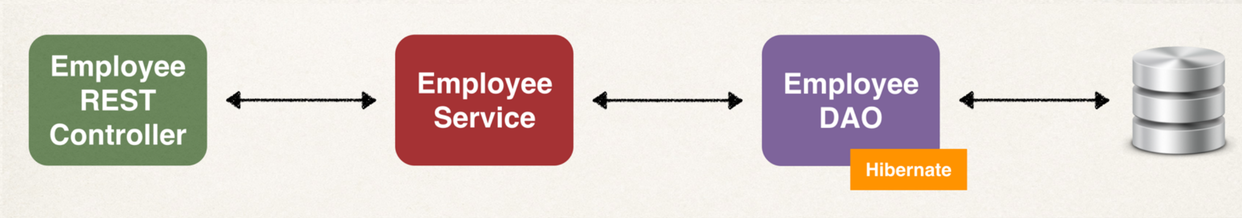
Rest Controller Methods - Find and Add

Development Process

Step-By-Step

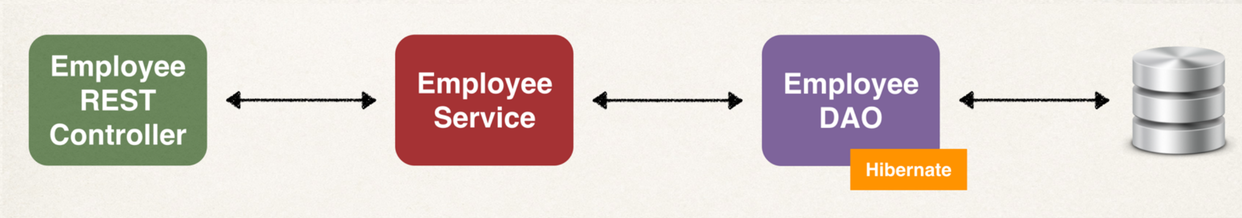
1. Set up Database Dev Environment
2. Create Spring Boot project using Spring Initializr
3. Get list of employees
4. Get single employee by ID
5. Add a new employee
6. Update an existing employee
7. Delete an existing employee



Development Process

Step-By-Step

1. Set up Database Dev Environment
2. Create Spring Boot project using Spring Initializr
3. Get list of employees
4. Get single employee by ID
5. Add a new employee
6. Update an existing employee
7. Delete an existing employee

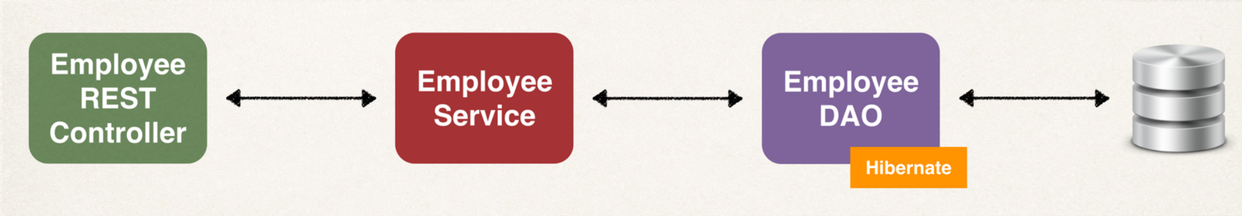


Development Process

Step-By-Step

1. Set up Database Dev Environment
2. Create Spring Boot project using Spring Initializr
3. Get list of employees
4. Get single employee by ID
5. Add a new employee
6. Update an existing employee
7. Delete an existing employee

Rest Controller methods



Real-Time Project

HTTP Method		CRUD Action
GET	/api/employees	Read a list of employees
GET	/api/employees/{employeeId}	Read a single employee
POST	/api/employees	Create a new employee
PUT	/api/employees	Update an existing employee
DELETE	/api/employees/{employeeId}	Delete an existing employee

Real-Time Project

Checkpoint

HTTP Method		CRUD Action
GET	/api/employees	Read a list of employees
GET	/api/employees/{employeeId}	Read a single employee
POST	/api/employees	Create a new employee
PUT	/api/employees	Update an existing employee
DELETE	/api/employees/{employeeId}	Delete an existing employee

Real-Time Project

Checkpoint

HTTP Method		CRUD Action
GET	/api/employees	Read a list of employees
GET	/api/employees/{employeeId}	Read a single employee
POST	/api/employees	Create a new employee
PUT	/api/employees	Update an existing employee
DELETE	/api/employees/{employeeId}	Delete an existing employee

Real-Time Project

Checkpoint

HTTP Method		CRUD Action
GET	/api/employees	Read a list of employees
GET	/api/employees/{employeeId}	Read a single employee
POST	/api/employees	Create a new employee
PUT	/api/employees	Update an existing employee
DELETE	/api/employees/{employeeId}	Delete an existing employee

Real-Time Project

Checkpoint

HTTP Method		CRUD Action
GET	/api/employees	Read a list of employees
GET	/api/employees/{employeeId}	Read a single employee
POST	/api/employees	Create a new employee
PUT	/api/employees	Update an existing employee
DELETE	/api/employees/{employeeId}	Delete an existing employee

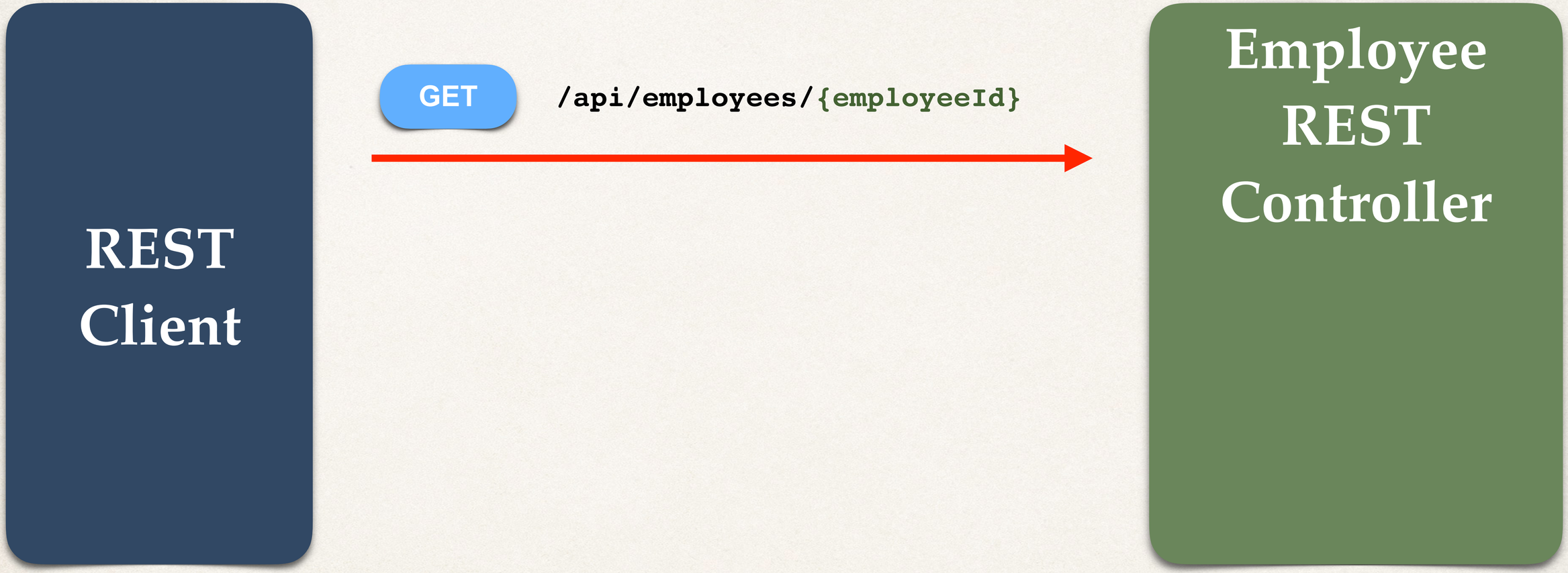
Read a Single Employee

Read a Single Employee

REST
Client

Read a Single Employee

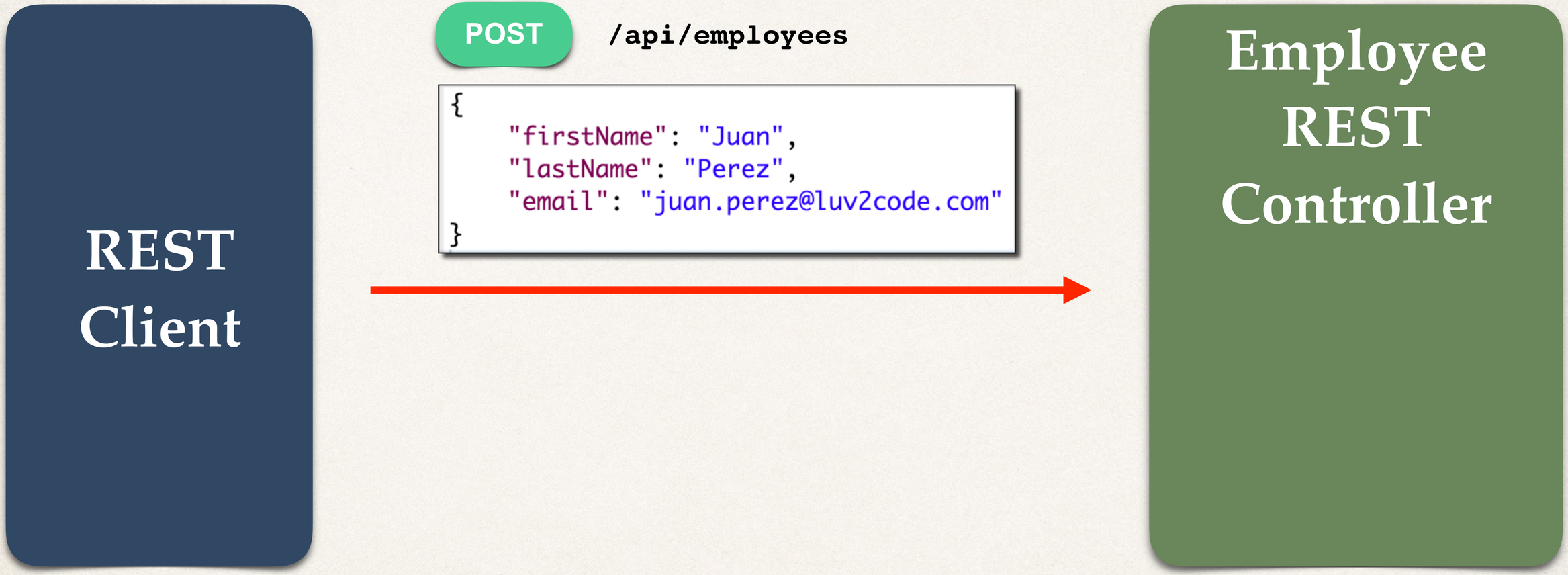
Read a Single Employee



Read a Single Employee

Create a New Employee

Create a New Employee



Create a New Employee

Since new employee, we are not passing id / primary key

REST Client

POST /api/employees

```
{
    "firstName": "Juan",
    "lastName": "Perez",
    "email": "juan.perez@luv2code.com"
}
```

Employee REST Controller

Create a New Employee

Since new employee, we are not passing id / primary key

REST Client

POST /api/employees

```
{
    "firstName": "Juan",
    "lastName": "Perez",
    "email": "juan.perez@luv2code.com"
}
```

```
{
    "id": 7,
    "firstName": "Juan",
    "lastName": "Perez",
    "email": "juan.perez@luv2code.com"
}
```

Employee REST Controller

Create a New Employee

Since new employee, we are not passing id / primary key

REST Client

POST /api/employees

```
{
    "firstName": "Juan",
    "lastName": "Perez",
    "email": "juan.perez@luv2code.com"
}
```

```
{
    "id": 7,
    "firstName": "Juan",
    "lastName": "Perez",
    "email": "juan.perez@luv2code.com"
}
```

Response contains new id / primary key

Employee REST Controller

Sending JSON to Spring REST Controllers

Sending JSON to Spring REST Controllers

- When sending JSON data to Spring REST Controllers

Sending JSON to Spring REST Controllers

- When sending JSON data to Spring REST Controllers

- For controller to process JSON data, need to set a HTTP request header

Sending JSON to Spring REST Controllers

- When sending JSON data to Spring REST Controllers

- For controller to process JSON data, need to set a HTTP request header
  - `Content-type: application/json`

Sending JSON to Spring REST Controllers

- When sending JSON data to Spring REST Controllers
- For controller to process JSON data, need to set a HTTP request header
  - `Content-type: application/json`
- Need to configure REST client to send the correct HTTP request header

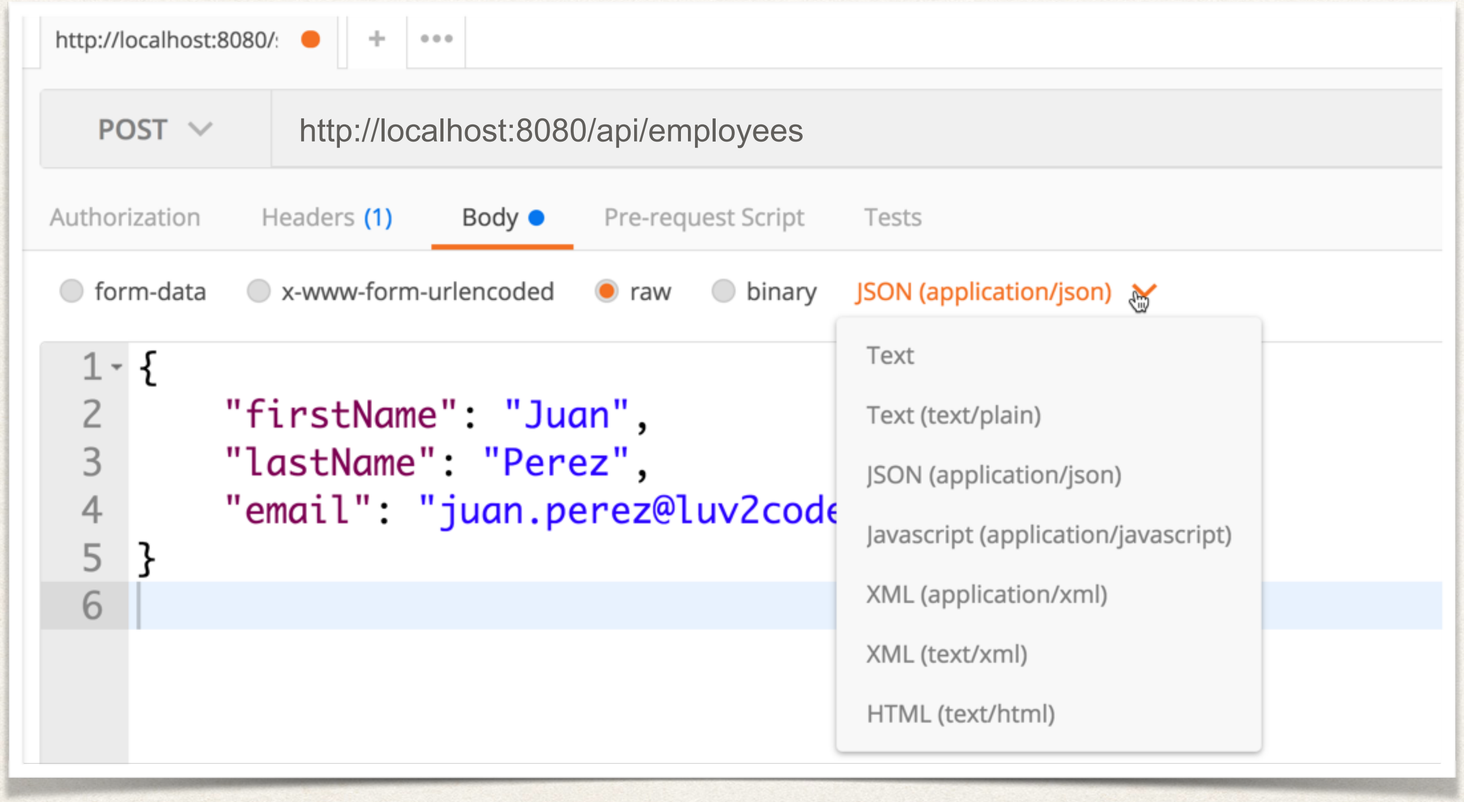
Postman - Sending JSON in Request Body

Postman - Sending JSON in Request Body

- Must set HTTP request header in Postman

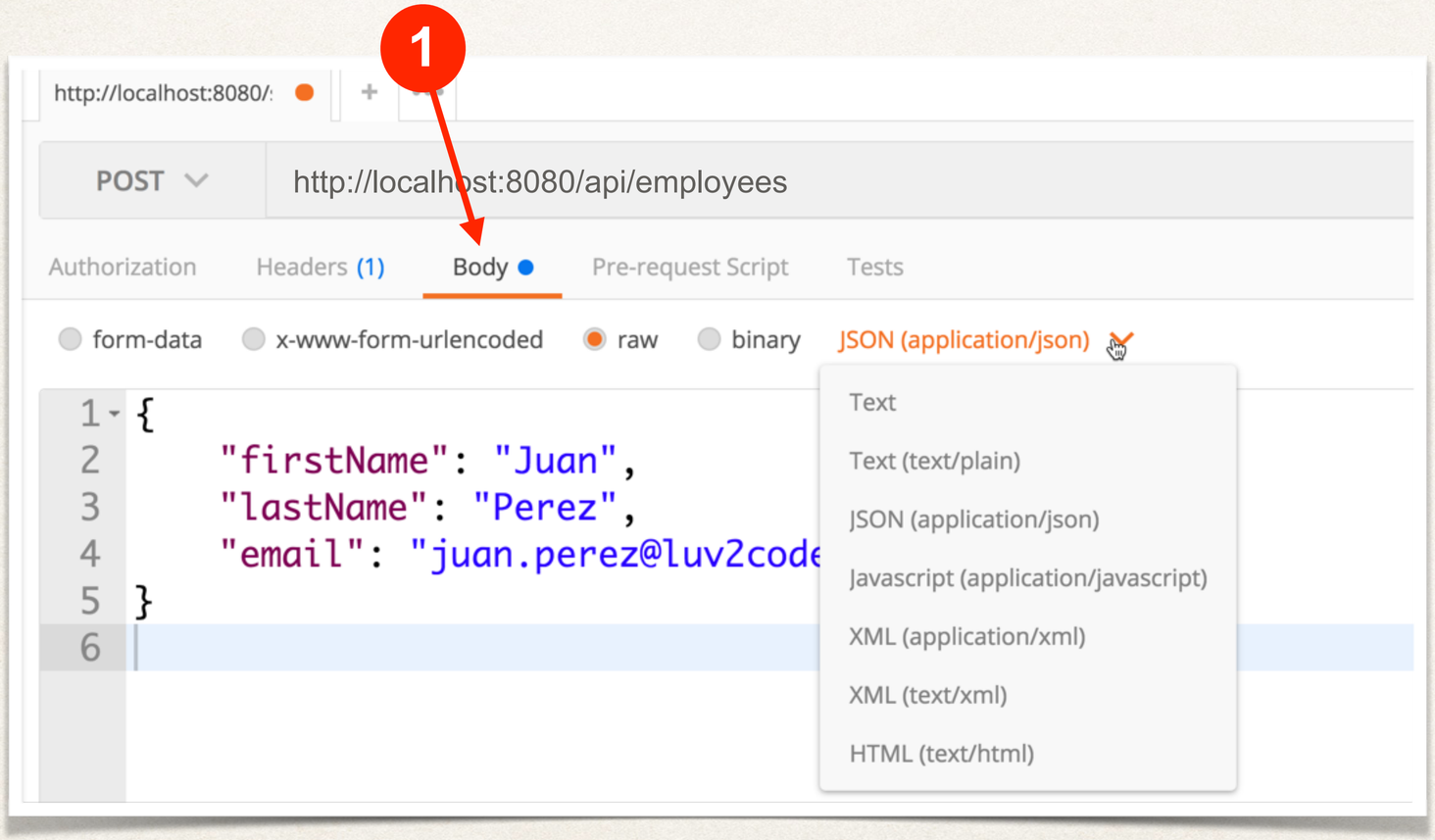
Postman - Sending JSON in Request Body

- Must set HTTP request header in Postman



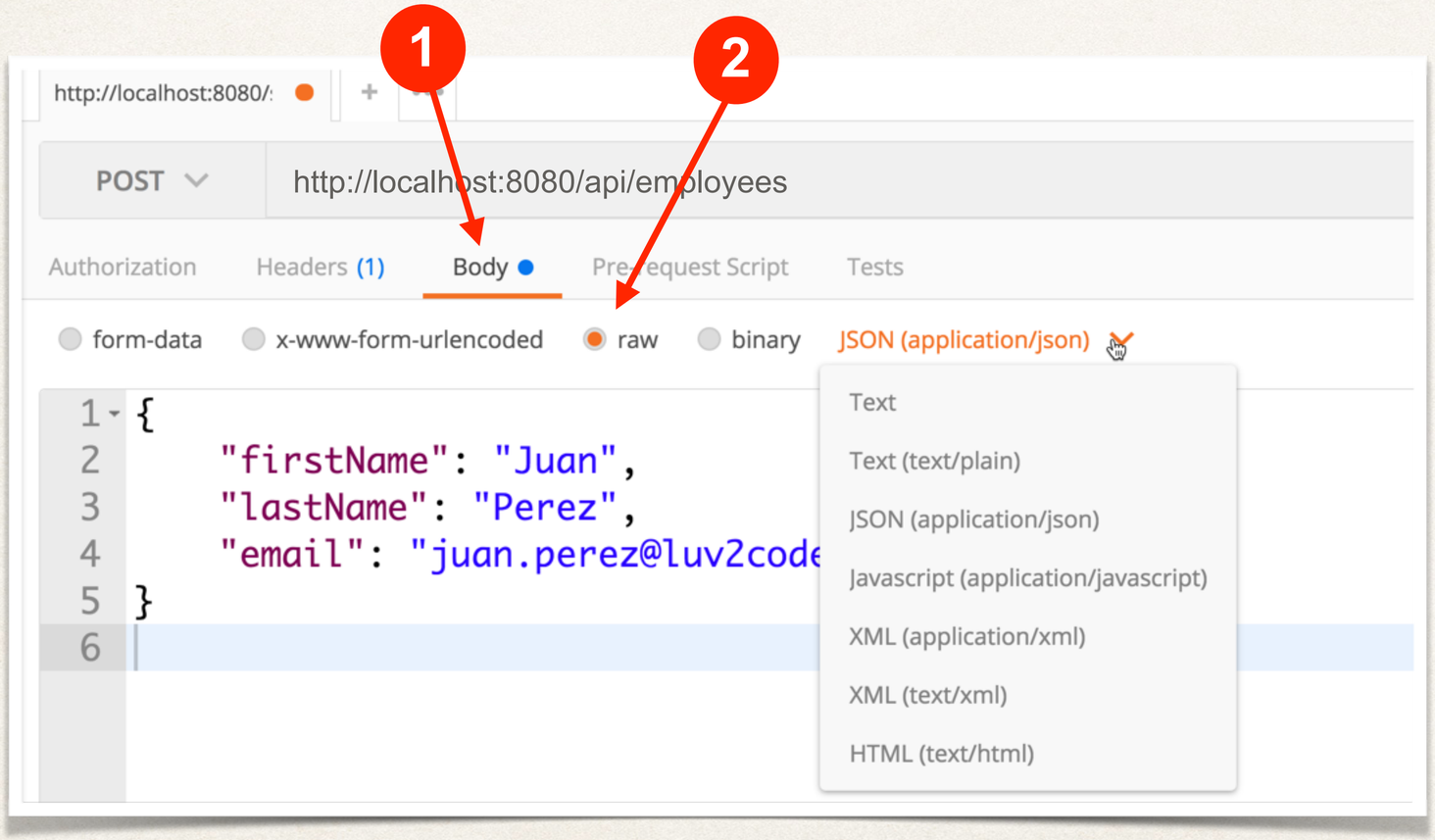
Postman - Sending JSON in Request Body

- Must set HTTP request header in Postman



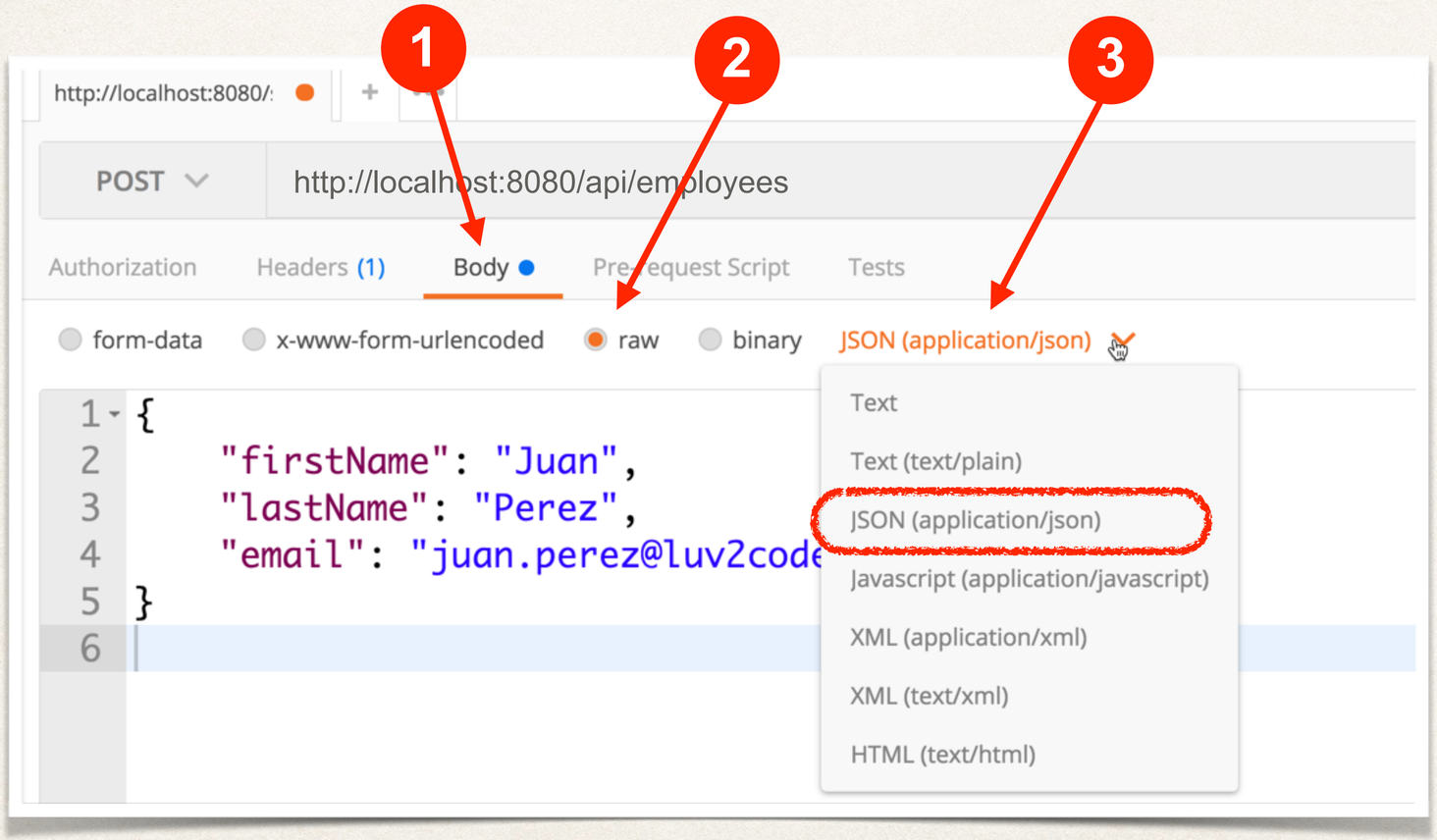
Postman - Sending JSON in Request Body

- Must set HTTP request header in Postman



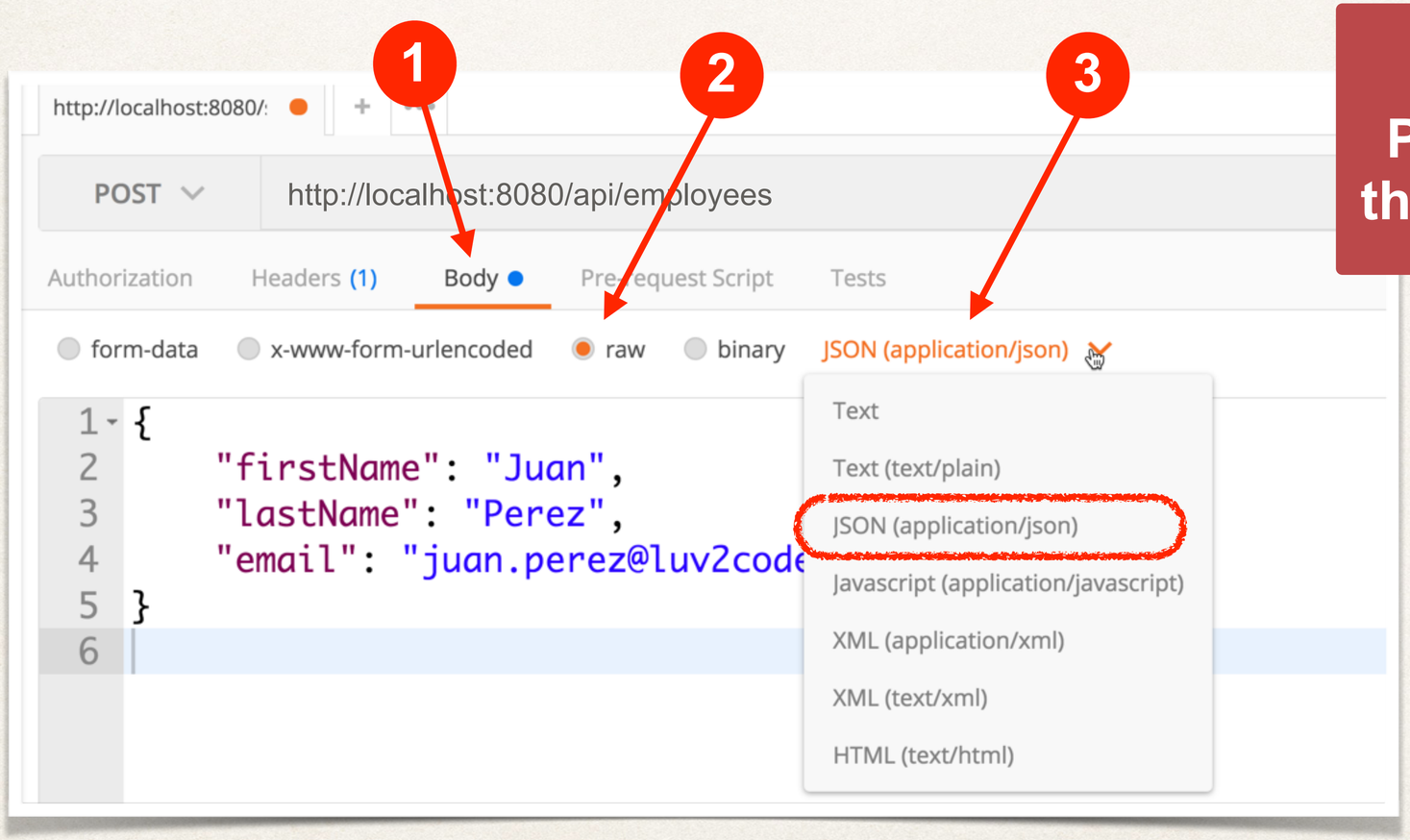
Postman - Sending JSON in Request Body

- Must set HTTP request header in Postman



Postman - Sending JSON in Request Body

- Must set HTTP request header in Postman



Based on these configs, Postman will automatically set the correct HTTP request header