

Step 4: Disable HTTP PATCH method

- Previously, we configured the Spring Data REST APIs for read-only
 - Disabled: HTTP GET, POST and DELETE
- However, we need to ALSO disable: HTTP PATCH


Step 4: Disable HTTP PATCH method

File: MyDataRestConfig.java

```
@Override
public void configureRepositoryRestConfiguration(RepositoryRestConfiguration config, CorsRegistry cors) {

    HttpMethod[] theUnsupportedActions = {HttpMethod.PUT, HttpMethod.POST, HttpMethod.DELETE, HttpMethod.PATCH};

    // disable HTTP methods for ProductCategory: PUT, POST, DELETE and PATCH
    disableHttpMethods(Product.class, config, theUnsupportedActions);
    disableHttpMethods(ProductCategory.class, config, theUnsupportedActions);
    disableHttpMethods(Country.class, config, theUnsupportedActions);
    disableHttpMethods(State.class, config, theUnsupportedActions);
    ...
}
```



Step 5: Modify Spring Data REST Detection Strategy

- By default, Spring Data REST will expose REST APIs for Spring Data Repositories
- This works in general ... but you may not want to expose all
- For example, in our case, we have CustomerRepository
 - BUT ... we don't want to expose this as a REST API
 - We only want to use it internally like for checking if customer email exists
- REST endpoint `/api/customers` is currently exposed let's fix this ASAP!

Step 5: Modify Spring Data REST Detection Strategy

- Spring Data REST has different detection strategies

Strategy	Description
ALL	Exposes all Spring Data repositories regardless of their Java visibility or annotation configuration
DEFAULT	Exposes public Spring Data repositories or ones explicitly annotated with <code>@RepositoryRestResource</code> and its exported attribute not set to false
VISIBILITY	Exposes only public Spring Data repositories regardless of annotation configuration
ANNOTATED	Only exposes Spring Data repositories explicitly annotated with <code>@RepositoryRestResource</code> and its exported attribute not set to false

Good approach for our app

Step 5: Modify Spring Data REST Detection Strategy

ANNOTATED

Only exposes Spring Data repositories explicitly annotated with `@RepositoryRestResource` and its `exported` attribute not set to `false`

File: StateRepository.java

```
@RepositoryRestResource(exported = true)
public interface StateRepository extends JpaRepository<State, Integer> {

    ... ..

}
```

Default value of `exported` is `true` ...
so we don't have to list it

File: StateRepository.java

```
@RepositoryRestResource
public interface StateRepository extends JpaRepository<State, Integer> {

    ... ..

}
```


Step 5: Modify Spring Data REST Detection Strategy

- Configure detection strategy

ANNOTATED

Only exposes Spring Data repositories explicitly annotated with `@RepositoryRestResource` and its exported attribute not set to false

File: application.properties

```
spring.data.rest.detection-strategy=ANNOTATED  
... ..
```