

Hiroperu's Life / Post Section / SpringでWeb APIのレスポンスJSON内のnullを空文字...

目次

シリアライザーの実装

Jackson2ObjectMapperBuilderの拡張
Jackson2ObjectMapperFactoryBeanの拡張
spring-rest-mvc.xmlの編集
おわりに
参考にさせてもらったサイト様

SpringでWeb APIのレスポンスJSON内のnullを空文字に変換してもらうようにした話

📅 2022/4/3 : 5分で読めます : 🌐 hiroperu
🏷️ #java #Spring #WebAPI #サーバーサイドアプリケーション

やりたいこと

Spring MVCでHttpMessageConverterを通してレスポンスボディに設定されるJSONで値が入っていない(null)際に空文字("")に自動で変換してもらいたい。
こんなイメージ

json

```
1 {
2   hoge: "hoge",
3   fuga: null
4 }
```

これを

json

```
1 {
2   hoge: "hoge",
3   fuga: ""
4 }
```

こうしたい。

fugaはnullなのですが、空文字("")にしたい。また、別のBeanをレスポンスにマッピングする際も同様の設定で同じような挙動をさせたい。
つまり、webAPI全体での挙動としたい。

🎯 読むとハッピーになれるかも知れない人

- JavaでSpringを使ってWebAPIを作っている人
- Jackson2HttpMessageConverter使ってレスポンスをマッピングんしている人
- nullを空文字に変換してレスポンス返したい人

設定するためにやること

大まかに次の通りです。

1. org.springframework.http.converter.json.Jackson2ObjectMapperBuilderの拡張
2. org.springframework.http.converter.json.Jackson2ObjectMapperFactoryBeanの拡張
3. NullValueSerializerクラスの実装
4. spring-rest-mvc.xmlの編集

spring-rest-mvc.xmlに以下の様に設定されていて、これらがレスポンスにBean(DTOともいう)をマッピングしているのδειじてやります。

xml

```
1 <bean id="jsonMessageConverter"
2     class="org.springframework.http.converter.json.MappingJackson2H
3     <property name="objectMapper" ref="objectMapper" />
4 </bean>
5
6 <bean id="objectMapper" class="jp.go.mlit.motas.etsuran.api.common.Jack
```

構造としてはMappingJackson2HttpMessageConverterが使うobjectMapperがJackson2ObjectMapperFactoryBeanとなっているので、Jackson2ObjectMapperFactoryBeanをいじって、独自実装したシリアライザーを差し込んでいきます。

シリアライザーの実装

nullを空文字に設定するシリアライザーを実装します。

java

```
1 import org.codehaus.jackson.JsonGenerator;
2 import org.codehaus.jackson.JsonProcessingException;
3 import org.codehaus.jackson.map.JsonSerializer;
4
5 public static class NullValueSerializer extends JsonSerializer<Object>
6 {
7     @Override
8     public void serialize(Object t, JsonGenerator jsonGenerator,
9     throws IOException, JsonProcessingException {
10         jsonGenerator.writeString("");
11     }
12 }
```

Jackson2ObjectMapperBuilderの拡張

拡張したいのは、configureメソッドです。このメソッドが呼び出し元となるJackson2ObjectMapperFactoryBeanから呼ばれて、実際にレスポンスにBeanをマッピングする際に利用するObjectMapperの設定を入れ込んでくれます。

次の様に拡張します。

java

```
1 import 実装したNullValueSerializer;
2
3 public class MyBuilder extends Jackson2ObjectMapperBuilder {
4
5     @Override
6     public void configure(ObjectMapper objectMapper){
7         super.configure(objectMapper);
8         DefaultSerializerProviderImpl dsp = new DefaultSerializerProv
9         dsp.setNullValueSerializer(new NullValueSerializer());
10        objectMapper.setSerializerProvider(dsp);
11    }
12 }
```

Jackson2ObjectMapperFactoryBeanの拡張

拡張したJackson2ObjectMapperBuilderを利用する様にこいつも書き換えます。気を付けなければならないのは、継承元のJackson2ObjectMapperFactoryBean側のメソッドは呼んでほしくないので、全てのメソッドをオーバーライドする必要があります。

親のメソッドを呼ばれたくない理由は、親クラスのメソッド内で利用されるbuilderは拡張前のJackson2ObjectMapperBuilderを利用しているからです。

java

```
1 public class MyMapperFactoryBean extends Jackson2ObjectMapperFactoryBean
2 {
3     private final MyBuilder builder = new MyBuilder();
4
5     @Nullable
6     private ObjectMapper objectMapper;
7
8     @Override
9     public void setObjectMapper(ObjectMapper objectMapper) {
10         this.objectMapper = objectMapper;
11     }
12
13     @Override
14     public void setCreateXmlMapper(boolean createXmlMapper) {
15         this.builder.createXmlMapper(createXmlMapper);
16     }
17
18     @Override
19     public void setFactory(JsonFactory factory) {
20         this.builder.factory(factory);
21     }
22
23     //中略
24
25     @Override
26     public void afterPropertiesSet() {
27         if (this.objectMapper != null) {
28             this.builder.configure(this.objectMapper);
29         }
30         else {
31             this.objectMapper = this.builder.build();
32         }
33     }
34
35     @Override
36     @Nullable
37     public ObjectMapper getObject() {
38         return this.objectMapper;
39     }
40
41     @Override
42     public Class<?> getObjectType() {
43         return (this.objectMapper != null ? this.objectMapper.get
44     }
45
46     @Override
47     public boolean isSingleton() {
48         return true;
49     }
50
51 }
```

afterPropertiesSetメソッドでJackson2ObjectMapperBuilderを拡張したconfigureが呼ばれて、NullValueSerializerがセットされます。

spring-rest-mvc.xmlの編集

最後に、Bean定義しているspring-rest-mvc.xmlに拡張してあげたクラスを使う様に書いてあげます。

spring-rest-mvc.xmlに以下の様に設定されていて、これらがレスポンスにBean(DTOともいう)をマッピングしているのδειじてやります。

xml

```
1 <bean id="jsonMessageConverter"
2     class="org.springframework.http.converter.json.MappingJackson2H
3     <property name="objectMapper" ref="objectMapper" />
4 </bean>
5
6 <bean id="objectMapper" class="MyMapperFactoryBean"></bean>
```

おわりに

かなり力技での設定になってしまったと思います。Jackson2ObjectMapperBuilderを見ると、

serializerというメソッドがあるのでこいつをBean定義しているxmlから呼び出してNullValueSerializerをセットできればもっとスマートに思っています。

ですが、今回はどのようなレスポンスボディの場合でも統一して空文字にしてほしかつたので、

JsonSerializer.handledType()に統一的なレスポンスDTOのオブジェクトとなるクラスを返す様にすることも考えたのですが、ちょっとうまくいかなかったです。

Object.classだとUnknown handled type と言われてエラーになってしまうのです。

この辺りもっとスマートにできるよってのがあればコメントいただけるととても嬉しいです。

参考にさせてもらったサイト様

[NullValueSerializerの実装まわり](#)[configureの拡張内容まわり](#)

共有



著者

hiroperu

子育て初心者

[🔍 クレバードリッパーは水出しコーヒーを作るときに使っても便利だよって...](#)[マニュアル式のエスプレッソマシン ROK EspressoGCを買った話 →](#)

Disqus comments not available by default when the website is previewed locally.

