

AULA 4

1. O sistema legado expõe serviços através do desenvolvimento de uma API e os mesmos ficam disponíveis para ser consumidos através da internet dessa forma sendo possível que o desenvolvedor de app possa construir um app para smartphone que consome serviços do sistema legado, publicando esse app na loja de aplicativos do smartphone que ficará disponível para o usuário final fazer o download e utilizar.
2. Está sendo criado um novo conceito para desenvolvimento intitulado de *Mobile First* onde se começa a planejar e desenvolver projetos web, desde um pequeno site até um grande sistema, primeiramente para dispositivos móveis e somente depois para desktops/notebooks. O design também precisará ser revisto utilizando conceitos de acessibilidade e usabilidade.
1. **Faça um resumo de 2 a 3 páginas desse guia com recomendações precisas. Para recomendação fornecida, dê um exemplo de uma API no contexto de cadastro de clientes.**

1. Nouns are good; verbs are bad

Mantenha sua URL base simples e intuitiva porque é a forma mais importante da sua API se comunicar sem a necessidade de uma documentação.

A primeira URL é para coleção e a segunda é para um elemento específico da coleção.

Ex.: /clientes /clientes/1234

Use os verbos do HTTP para fazer as suas requisições da URL

2. Plural nouns and concrete names

Você deve escolher os substantivos singular ou no plural para seus nomes? Tendo em conta que a primeira coisa que a maioria das pessoas provavelmente fazer com uma API RESTful é um GET, pensamos que é mais intuitivo de usar os substantivos no plural. Mas, acima de tudo, evitar um modelo misto em que você usa singular para alguns recursos, plural para os outros. Ser coerente permite que os desenvolvedores prever e adivinhar as chamadas de método da sua API.

3. Simplify associations - sweep complexity under the ‘?’

Recursos quase sempre têm relacionamentos com outros recursos. Então estamos usando verbos HTTP para operar sobre os recursos e coleções. Nossos clientes pertencem a algum produto. Para obter todos os clientes que pertencem a um produto específico, ou para criar um novo cliente para esse produto, fazer um GET ou POST:

GET /products/5678/clients

POST /products/5678/clients

Make it simple for developers to use the base URL by putting optional states and attributes behind the HTTP question mark. To get all clients that bought more than \$200 on the date of 2016/05/01:

Torne simples para os desenvolvedores que utilizam a URL base, colocando estados e atributos opcionais atrás do ponto de interrogação HTTP.

Para obter todos os clientes que compraram mais de R\$200 na data de 2016/05/01:

GET /clients?date=20160501&value=200

4. Handling errors

Muitos desenvolvedores de software, nem sempre gostam de pensar sobre exceções e tratamento de erros, mas é uma peça muito importante do quebra-cabeça para qualquer desenvolvedor de software, e especialmente para os designers de API.

5. Tips for versioning

Controle de versão é uma das considerações mais importantes ao projetar seu Web API. Nunca lançar uma API sem uma versão e fazer a versão obrigatória.

6. Pagination and partial response

Resposta parcial permite dar aos desenvolvedores apenas a informação de que eles precisam. Tome por exemplo, um pedido de um tweet sobre a API do Twitter. Você terá muito mais do que um aplicativo de Twitter típico muitas vezes precisa - incluindo o nome da pessoa, o texto do *tweet*, um *timestamp*, quantas vezes a mensagem foi *re-tweeted*, e um monte de metadados.

7. Supporting multiple formats

Recomendamos que você mantenha mais de um formato. Geralmente, é possível automatizar o mapeamento do formato para o formato.

8. What about attribute names?

Use JSON como padrão

Use capitalização média (aka CamelCase)

Use maiúscula ou minúscula dependendo do tipo do objeto

9. Tips for search

Pesquisa Global

/search?q=fluffy+fur

Pesquisa no Escopo

/owners/5678/dogs?q=fluffy+fur

Resultados formatados

/search.xml?q=fluffy+fur

10. Authentication

Use o mais recente OAuth - OAuth 2.0. Isso significa que aplicativos Web ou móveis que expõem APIs não tem que compartilhar senhas. Ele permite que o provedor de API cancele os tokens para um usuário individual, para um aplicativo inteiro, sem exigir que o usuário altere sua senha original. Isto é crítico se um dispositivo móvel é comprometido ou se um aplicativo não autorizado é descoberto.

11. Chatty APIs

Ao projetar sua API e recursos, tente imaginar como os programadores vão usá-lo para construir uma interface de usuário, um aplicativo para iPhone, ou muitos outros projetos apps. Em algumas API apenas para construir uma interface de usuário simples ou aplicativo, você tem dezenas ou centenas de API de volta para o servidor.

Exemplo do Estilo de APIs com o Spring Boot

1. Rode o exemplo fornecido pelo professor do projeto SpringBoot dentro do ambiente Eclipse. Explique o código fonte do projeto em suas palavras (10 a 20 linhas).

Antes de ser executado, o script é pré-processado pelo Spring Boot tirando máximo proveito das características dinâmicas da linguagem Groovy, que permite modificar código em tempo de execução através do seu *Meta Object Protocol* (MOP) e em

tempo de compilação usando transformações AST, capazes de, dentre outras coisas, incluir instruções de import ausentes, assim como adicionar anotações em alguns dos métodos. Outra característica do Groovy bastante explorada pelo Boot é o *Grape*, gerenciador de dependências nativo da linguagem, similar ao Maven ou Ivy. Com ele o desenvolvedor obtém de forma transparente em tempo de execução todas as dependências necessárias para a execução do script como, por exemplo, drivers JDBC, bibliotecas, etc.

Quando executamos o comando *spring run*, o Spring Boot inicia o processo de análise do código fonte do script Groovy que passamos como parâmetro.

Ao encontrar a anotação `@RestController`(pacote `org.springframework.web.bind.annotation`), o script será interpretado como a implementação de um serviço REST.

2. Qual o papel arquitetural do SpringBoot? Que tecnologias de outros fornecedores são similares a este framework?

O projeto Spring Boot nos permite criar dois tipos de aplicações: as tradicionais, escritas primariamente em Java; e uma segunda forma, chamada pela equipe de desenvolvimento do Boot de “Spring Scripts”, que nada mais são do que códigos escritos em Groovy – o que não é de se estranhar, dado que desde o lançamento da versão 4.0 do framework há uma forte tendência da Pivotal em abraçar esta linguagem de programação e torná-la cada vez mais presente no dia a dia do programador Java.

Frameworks como Ruby on Rails e Grails já o aplicam há bastante tempo

3. Desenhe um diagrama de componentes da aplicação executada no passo 4.

