Especialista Spring Boot

Como funciona?

Especialista Spring Boot

Spring Boot -Como funciona?

Módulo WEB

Starter WEB

API's REST - Aplicações WEB

Módulo Spring Data

Starter Data JPA

JPA - Acesso a Dados - Hibernate, etc

Módulo Security

Starter Security

Spring Security - OAuth2 - JWT, etc

Módulo Testes

Starter Test

JUnit - Mockito, etc

Módulo Validação

Starter Validation

Bean Validation

Spring Framework

Framework para aplicações Complexas

Injeção de Dependências (DI), Inversão de Controle (IOC)

Configurations

Configurações através de annotations

application.yml - application.properties

Beans

Componentes da Aplicação

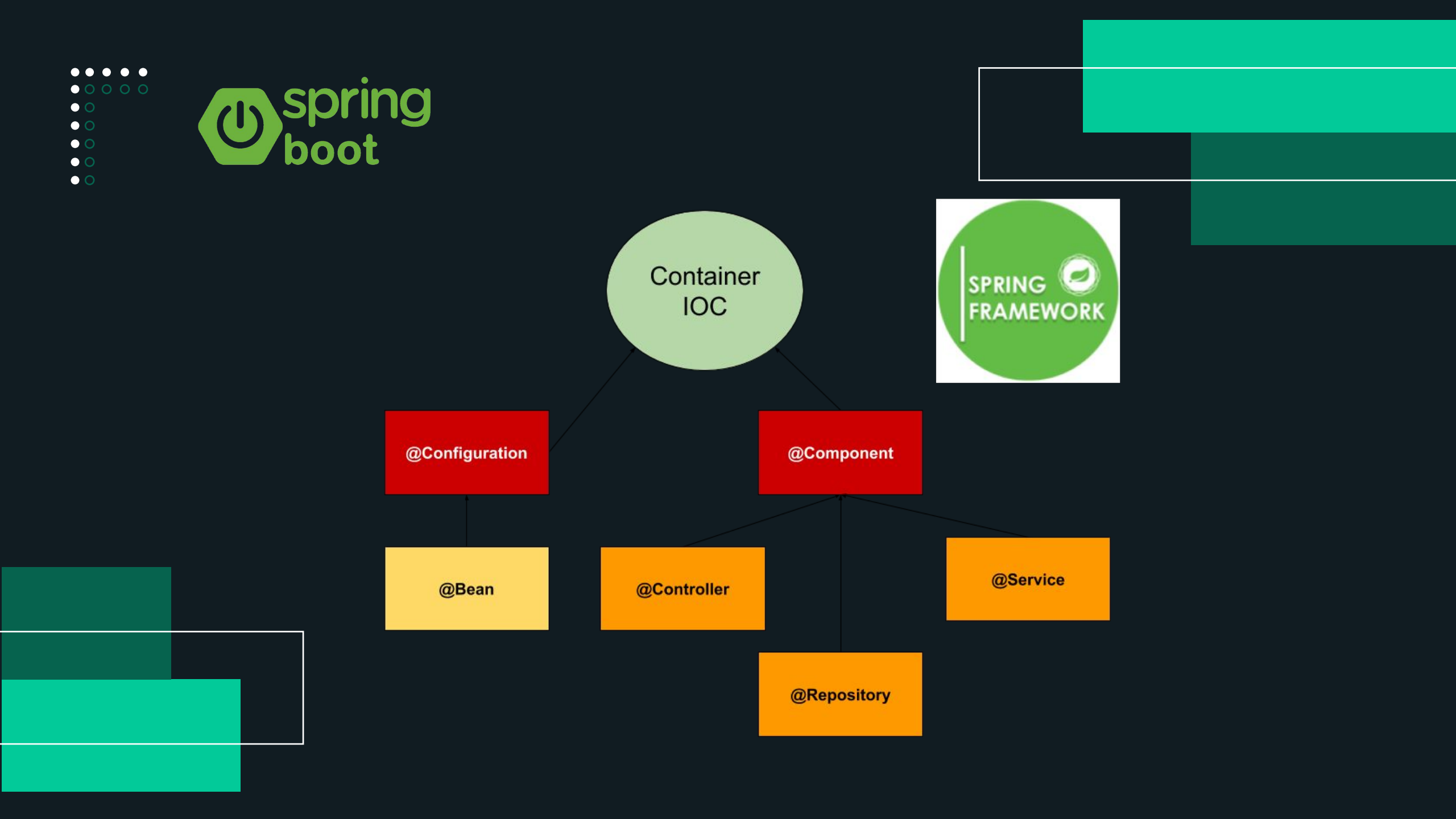
Definição de objetos de configuração

Arquitetura Spring

- Container Spring
- Componentes Spring
- Application Context
- Scan de Componentes
- Configurações (Configurations) e Beans
- application.yml/properties e Profiles

Spring Framework e Spring Boot

- Ecossistema Spring
- Spring Data
- Spring Security
- Spring Cloud
- Spring Rest


spring boot
Container IOC
SPRING FRAMEWORK
@Configuration
@Component
@Bean
@Controller
@Service
@Repository

Insira o texto

PREENCHA AQUI O TEXTO DO TÍTULO DO

Insira o texto do

Obrigado por usar nosso modelo PPT, digite o conteúdo de texto que você precisa aqui, obrigado por usar nosso modelo PPT. Obrigado por usar nosso modelo PPT, digite o conteúdo de texto que você precisa aqui, obrigado por usar nosso modelo PPT.

Módulo WEB

Starter WEB

Obrigado por usar nosso modelo PPT

Módulo Spring Data

Starter Data JPA

Obrigado por usar nosso modelo PPT

Módulo Security

Starter Security

Obrigado por usar nosso modelo PPT

Especialista Spring Boot

API REST

PARTE 01

Protocolo HTTP

Protocolo HTTP

Hypertext Transfer Protocol

- Baseado em texto
- Sem estado
- Cliente-Servidor
- Modelo Request-Response
- HTTPS (HTTP Secure)

Estrutura de uma Requisição

- URL
- Cabeçalho (Header)
- Corpo (Body - Opcional)
- Método

Requisição (Request)

Método: POST

URL: http://api.com/clientes

Headers:

- Authorization: Bearer
- Content-Type: application/json

Body:

{ “nome”: “Zezinho”, “cpf”: “0123456790” }

Estrutura de uma Resposta

- Cabeçalho (Header)
- Corpo (Body - Opcional)
- Código de Status

Resposta (Response)

Headers:

- Content-Type: text/html
- Content-length: 342

Status Code: 200 (ok)

Body:

```
<html>
   <head></head>
  <body>
     <h1>hello</h1>
  </body>
</html>
```

Request Headers

- Accept
- Content-Type
- Authorization
- Host
- User-Agent
- Referer

Response Headers

- Content-Type
- Content-length
- Set-Cookie
- Cache-Control
- Location

Request e Response Body

- JSON (application/json)
- Texto (text/plain)
- XML (application/xml)
- HTML (text/html)
- JPEG (image/jpg)
- form-urlencoded

Métodos/Verbos HTTP

POST

Criar Recursos, Enviar Dados

GET

Obter Recursos

PUT

Atualizar Recursos

DELETE

Excluir Recursos

PATCH

Atualização Parcial

HEAD

Verificação

OPTIONS

Descreve opções de comunicação

Códigos de Status Response

100

Informativos

200

200 - OK
201 - Created
202 - Accepted
204 - No Content

300

Redirecionamentos

400

400 - Bad Request
401 - Unauthorized
403 - Forbidden
404 - Not found
405 - Method not allowed
422 - Unprocessable entity

500

Erros no servidor

PARTE 02

Modelando API's RESTful

REST

Representational State Transfer

- Princípios para projetar serviços em rede distribuídos
- Interface para uma comunicação padronizada
- RESTful?

Identificação de Recursos

- Utilize substantivos no plural, não verbos

Certo:
POST - http://api.com/clientes
PUT - http://api.com/clientes

Errado:
POST - http://api.com/salvar-cliente
PUT- http://api.com/cliente/atualizar

Exemplo Identificação de Recursos RESTful

POST - /clientes
PUT - /clientes
GET - /clientes/48891
GET - /clientes?nome-inicial=Maria
DELETE - /clientes/25874
PATCH - /clientes/5456

Subrecursos

- Utilize substantivos no plural, não verbos

Certo:
GET - http://api.com/clientes/565/dependentes
GET - http://api.com/produtos/154/categorias
POST - http://api.com/clientes/telefones
PUT- http://api.com/vendas/366/itens

Requisição POST

- Verbo: POST
- URL: /produtos
- Headers:
  - Accept: “application/json”
  - Content-type: “application/json”
- Body: {
  “nome”: “HD 1tb XPTO”,
  “preco”: 250.0
  }

Response POST

- Código de Status:
  - 201 - CREATED
  - 202 - ACCEPTED
- Headers:
  - Location: “http://api.com/produtos/987777
  - Content-Type: “application/json”

Erros

- Código de Status:
  - 400 ou 422 - Erro Validação
  - 500 - Erro de server
- Headers:
  - Content-Type: “application/json”
- Body: {
  “message”: “Erro validação”
  “status”: 400,
  “detail: “nome não pode ser nulo”
  }

Requisição PUT

- Verbo: PUT
- URL: /produtos/6548711
- Headers:
  - Content-type: “application/json”
- Body: {
  “nome”: “HD 1tb XPTO”,
  “preco”: 260.0
  }

Response PUT

- Código de Status Sucesso:
  - 204 - No Content
  - 200 - OK

- Código de Status Erro:
  - 404 - Not Found
  - 400 - Bad Request
  - 422 - Unprocessable Entity

IDEMPOTENTE - Mesmo Resultado

Requisição GET (Recurso Específico)

- Verbo: GET
- URL: /produtos/6548711
- Headers:
  - Accept: “application/json”
- Body: NÃO ACEITA

Requisição GET (Pesquisa)

- Verbo: GET
- URL: /produtos
- Query Params: ?nome=HD&preco-max=300
  - ex: http://api.com/produtos?nome=HD
- Headers:
  - Accept: “application/json”
- Body: NÃO ACEITA

Response GET

- Código de Status Sucesso:
  - 200 - OK
- Código de Status Erro
  - 404 - NOT FOUND (recurso específico)
- Headers:
  - Content-Type: “application/json”
- Body: [{
  “id”: 199487
  “nome”: “HD 1tb XPTO”,
  “preco”: 260.0,
  “categoria”: “Eletrônicos Informática”
  }]

Requisição DELETE (Recurso Específico)

- Verbo: DELETE
- URL: /produtos/6548711
- Headers:
  - Authorization: “Basic _ads52fa1sdf2a1sd2f3a10abd”
- Body: NÃO ACEITA

Requisição DELETE (Batch)

- Verbo: DELETE
- URL: /produtos
- Query Params: ?nome=HD&preco-max=300
  - ex: http://api.com/produtos?nome=HD
- Headers:
  - Authorization: “Bearer _asdf41a2sd3f4a5sd41fa24d”
- Body: NÃO ACEITA

Response DELETE

- Código de Status Sucesso:
  - 204 - No Content
- Código de Status Erro
  - 404 - NOT FOUND (recurso específico)

IDEMPOTENTE?