

---

---

# Desenvolvimento WEB

-

Full Stack Completo: Java + React

---

---

# Envio de e-mails com Templates Thymeleaf



+



# Envio de E-mails com Templates Thymeleaf

Implemente a seguinte funcionalidade:

sempre que houver um cadastro de um novo cliente no sistema, envie um email de boas vindas a ele.

# Envio de E-mails com Templates Thymeleaf

- Primeiro vamos adicionar um campo na entidade `Cliente` para armazenar o email:

```
...  
  
public class Cliente extends EntidadeAuditavel {  
  
    ...  
  
    @Column  
    private String email;  
  
    ...  
  
}
```

# Envio de E-mails com Templates Thymeleaf

- Depois, adicione o campo na classe do `ClienteRequest`:

```
...
public class ClienteRequest {

    ...

    @NotBlank(message = "O Email é de preenchimento obrigatório")
    @Email
    private String email;

    ...

    public Cliente buildCliente() {

        return Cliente.builder()
            .chaveEmpresa(chaveEmpresa)
            .nome(nome)
            .email(email)
            .cpf(cpf)
            .fone(fone)
            .foneAlternativo(foneAlternativo)
            .build();

    }
}
```

# Envio de E-mails com Templates Thymeleaf

- Agora vamos começar a implementar o código que envia o e-mail após o cadastro do cliente.
- Acrescente as dependências abaixo no arquivo `pom.xml`:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-mail</artifactId>
</dependency>

<dependency>
  <groupId>ognl</groupId>
  <artifactId>ognl</artifactId>
  <version>3.3.4</version>
</dependency>
```

# Envio de E-mails com Templates Thymeleaf

- Crie uma nova pasta dentro de `src -> main -> resources` chamada `templates`
- Dentro da pasta `templates`, crie o arquivo `bem_vindo_cliente.html`

# Envio de E-mails com Templates Thymeleaf

```
<!DOCTYPE html>
<html lang="pt-BR" dir="ltr" xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org" xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:o="urn:schemas-microsoft-com:office:office">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1 user-scalable=yes">
  <meta name="format-detection" content="telephone=no, date=no, address=no, email=no, url=no">
  <meta name="x-apple-disable-message-reformatting">
  <!--[if mso]>
  <noscript>
    <xml>
      <o:OfficeDocumentSettings>
        <o:PixelsPerInch>96</o:PixelsPerInch>
      </o:OfficeDocumentSettings>
    </xml>
  </noscript>
  <![endif]-->
  <style>
    @media only screen and (max-width: 600px){
      .watermark{
        background:#ffffff !important;
      }
    }
  </style>
</head>

<body class="body" style="margin: 0; padding: 0; background:#f9f9f9; font-family: 'Open Sans', 'Tahoma', sans-serif; line-height: 2.1em;">
  <div role="article" aria-roledescription="email" lang="pt-BR" dir="ltr" style="font-size:medium; font-size:max(16px, 1rem); font-family: 'Open Sans', 'Tahoma', sans-serif;
line-height: 2.1em;">
    <table role="presentation" style="width: 60.5em; max-width:100%; background:#ffffff; margin: 0 auto;" align="center" border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td>
```

Código do arquivo

bem\_vindo\_cliente.html em:



```
</br>

<table role="presentation" style="width:100%; background:#ffffff; margin: 0 auto;" align="center" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td style="padding: 0 1.5em;">
            <hr style="border-width: 0; background: #c0c0c0; color: #c0c0c0; height:.15em">
        </td>
    </tr>
</table>
<table role="presentation" style="width:100%; background:#ffffff; margin: 0 auto;" align="center" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td style="padding: 0 1.5em;">
            <h1 style="text-align: center; font-size: 1.5em; margin-bottom: 1.5rem;">Bem vindo ao nosso aplicativo!</h1>
        </td>
    </tr>
</table>
<table role="presentation" style="width:100%; background:#ffffff; margin: 0 auto;" align="center" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td style="padding: 0 1.5em;">
            <hr style="border-width: 0; background: #c0c0c0; color: #c0c0c0; height:.15em">
        </td>
    </tr>
</table>
<table role="presentation" style="width:100%; background:#ffffff; margin: 0 auto;" align="center" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td style="padding: 0 1.5em;">
            <p>Prezado <span th:text="{cliente.nome} ? : ''"></span></p>
            <p>Seu cadastro foi realizado com <b>sucesso</b>, seja bem-vindo ao nosso Aplicativo!</p>
            <p>Agora é muito mais fácil e seguro realizar pedidos, seja pelo computador ou celular.<br>
            Conheça nossos produtos e desfrute da melhor relação qualidade/preço do mercado!</p>
        </td>
    </tr>
</table>
<table role="presentation" style="line-height: 1.3rem; font-size: 1rem; font-style: italic; color: #5a5a5a; width:100%; margin: 0 auto;" align="center"
border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td style="padding: 2em 1.5em 1em">
            <p>Atenciosamente,<br> <span>Equipe Oxe Food - IFPE Campus Jaboatão</span></p>
        </td>
    </tr>
</table>
</td>
</tr>
</table>
</div>
</body>
</html>
```

# Envio de E-mails com Templates Thymeleaf

- Crie um novo pacote dentro de `br -> com -> ifpe -> oxefood -> modelo` chamado `mensagens`
- Dentro do pacote `mensagens`, crie a classe `EmailService.java`

# Envio de E-mails com Templates Thymeleaf

- Código do arquivo `EmailService.java` em:

```
package br.com.ifpe.oxefood.modelo.mensagens;

import java.nio.charset.StandardCharsets;
import java.util.Properties;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.JavaMailSenderImpl;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.scheduling.annotation.Async;
import org.springframework.stereotype.Component;
import org.thymeleaf.TemplateEngine;
import org.thymeleaf.context.Context;
import org.thymeleaf.templatemode.TemplateMode;
import org.thymeleaf.templateresolver.ClassLoaderTemplateResolver;

import br.com.ifpe.oxefood.modelo.cliente.Cliente;
import jakarta.mail.MessagingException;
import jakarta.mail.internet.InternetAddress;
import jakarta.mail.internet.MimeMessage;

/**
 * @author Roberto Alencar
 */
@Component
public class EmailService {
```

```
@Value("${spring.mail.username}")
String username;

@Value("${spring.mail.password}")
String password;

@Value("${spring.mail.host}")
String host;

@Value("${spring.mail.port}")
int port;

@Value("${spring.mail.properties.mail.smtp.auth}")
String smtpAuth;

@Value("${spring.mail.properties.mail.smtp.starttls.enable}")
String starttls;

private JavaMailSender emailSender;
```

Continua no próximo slide

```

public void enviarEmailConfirmacaoCadastroCliente(Cliente cliente) {

    String assuntoEmail = "Bem vindo ao nosso aplicativo";

    Context params = new Context();
    params.setVariable("cliente", cliente);

    this.sendMailTemplate("bem_vindo_cliente.html", cliente.getUsuario().getUsername(),
assuntoEmail, params);
}

@Async
private void sendMailTemplate(String template, String to, String subject, Context
params) {

    TemplateEngine templateEngine = new TemplateEngine();

    ClassLoaderTemplateResolver templateResolver = new ClassLoaderTemplateResolver();
    templateResolver.setPrefix("templates/");
    templateResolver.setSuffix(".html");
    templateResolver.setTemplateMode(TemplateMode.HTML);
    templateResolver.setCharacterEncoding("UTF-8");
    templateResolver.setOrder(0);

    templateEngine.setTemplateResolver(templateResolver);

    String content = templateEngine.process(template, params);
    this.sendMail(to, subject, content, Boolean.TRUE);
}

@Async
private void sendMail(String to, String subject, String content, Boolean html) {

    emailSender = getJavaMailSender();

    MimeMessage message = emailSender.createMimeMessage();

```

```

MimeMessageHelper helper = new MimeMessageHelper(message);

    try {

        helper.setFrom(new InternetAddress("not.reply@delifacil.com.br"));
        helper.setTo(to);
        helper.setSubject(subject);
        helper.setText(new String(content.getBytes(), StandardCharsets.ISO_8859_1), html);
        helper.setEncodeFileNames(true);

    } catch (MessagingException e) {
        e.printStackTrace();
    }

    emailSender.send(message);
}

private JavaMailSender getJavaMailSender() {

    JavaMailSenderImpl mailSender = new JavaMailSenderImpl();
    mailSender.setHost(host);
    mailSender.setPort(port);

    mailSender.setUsername(username);
    mailSender.setPassword(password);

    Properties props = mailSender.getJavaMailProperties();
    props.put("mail.transport.protocol", "smtp");
    props.put("mail.smtp.auth", smtpAuth);
    props.put("mail.smtp.starttls.enable", starttls);
    props.put("mail.debug", "false");
    props.put("spring.mail.properties.mail.smtp.starttls.enable", "true");

    return mailSender;
}
}

```

# Envio de E-mails com Templates Thymeleaf

- Acrescente no `application.properties` os dados de configuração da conta de e-mail que será utilizada para enviar os e-mails do sistema:

```
# Dados para envios de email

spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=seumail@ifpe.edu.br
spring.mail.password=suasenha
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

- OBS.: as configurações de exemplo acima é para o caso de se utilizar uma conta do gmail

# Envio de E-mails com Templates Thymeleaf

- Modifique o método `save` na classe `ClienteService` acrescentando a chamada para o método que envia o email na classe `EmailService`:

```
@Service
public class ClienteService {

    @Autowired
    private EmailService emailService;

    @Transactional
    public Cliente save(Cliente cliente) {

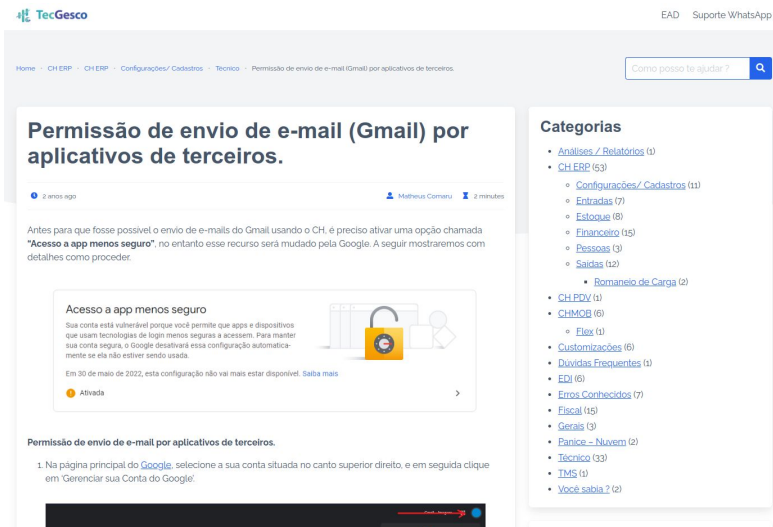
        cliente.setHabilitado(Boolean.TRUE);
        cliente.setVersao(1L);
        cliente.setDataCriacao(LocalDate.now());
        Cliente clienteSalvo = repository.save(cliente);

        emailService.enviarEmailConfirmacaoCadastroCliente(clienteSalvo);

        return clienteSalvo;
    }
}
```

# Envio de E-mails com Templates Thymeleaf

- **Atenção**, se for usar uma conta do Gmail para envio dos emails, é necessário dar a permissão de envio de e-mail (Gmail) por aplicativos de terceiros:
- Ver tutorial neste link:
  - <https://tecgesco.com/blog/permissoao-de-envio-de-e-mail-gmail-por-aplicativos-de-terceiros/>



# Envio de E-mails com Templates Thymeleaf

- Faça o teste do cadastro do cliente no Postman:

POST cliente - save

Oxe Food / Cliente / cliente - save

POST http://localhost:8080/api/cliente

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "chaveEmpresa": "123",
3   "email": "rroberto.alencar@gmail.com",
4   "nome": "João Pedro 19",
5   "cpf": "09567711417",
6   "fone": "81 98274.0284",
7   "foneAlternativo": "81 3038.1582"
8 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 22,
3   "email": "rroberto.alencar@gmail.com",
4   "enderecos": null,
5   "nome": "João Pedro 19",
6   "cpf": "09567711417",
7   "fone": "81 98274.0284",
8   "foneAlternativo": "81 3038.1582"
9 }
```

Bem vindo ao nosso aplicativo

Caixa de entrada

roberto.alencar@jaboatao.ifpe.edu.br

19:56 (há 0 minuto)

Bem vindo ao nosso aplicativo!

Prezado João Pedro 19

Seu cadastro foi realizado com **sucesso**, seja bem-vindo ao nosso Aplicativo!

Agora é muito mais fácil e seguro realizar pedidos, seja pelo computador ou celular.

Conheça nossos produtos e desfrute da melhor relação qualidade/preço do mercado!

Atenciosamente,  
Equipe Oxe Food - IFPE Campus Jaboatão

Responder Encaminhar



# Dúvidas



# Exercício

Implemente:

- O envio de e-mail de boas vindas no cadastro **do cliente**;
- O envio de e-mail de boas vindas no cadastro **de entregador**;



The image features a white background with two large, solid green abstract shapes. One shape is a semi-circle on the left side, and the other is a more complex, organic shape on the right side. Centered between these shapes is the text "Obrigado !".

Obrigado !