

Pontos de Impacto de customizações

5.50 Release 05



Olá, tudo bem?

Este é um **documento muito importante** que resume os pontos de impacto relacionados exclusivamente às customizações. Não deixe de ler todos os itens que estão no e-mail do Release Notes e no Inside.

Sumário

- Log4j – Customização
- Alteração do tipo do campo código de usuário
- Alteração do tamanho dos campos nos modelos
- Mudança na utilização de documentos estáticos
- Atualização da versão do Spring Boot para 2.6.6
- Nova versão do driver do MySql
- DevOps (clientes que utilizam esse recurso)
- Sequência de atividades para atualização
- Links úteis

Comentários PSP

- Caso o cliente só tenha 1 projeto no GIT para os dois ambientes, é importante que todas as customizações estejam homologadas para subir em produção, já que no fim da atualização os ambientes estarão iguais em relação às customizações (robôs, integrações e classes). Se o cliente tiver projetos separados, pode-se optar por fazer todas as mudanças e testes 2 vezes, uma para cada ambiente.

* O indicado é igualar as customizações, pois a atualização será mais rápida e eficiente.

Log4j – Customização



A partir de agora, a configuração de log não deve ser feita diretamente no arquivo log4j.properties.

Para simplificar, a configuração do log deverá ser realizada através “/system” ([URL_DO_AMBIENTE]/bpm/system/login)

Dessa forma, após a atualização para esse release, a configuração de log voltará para o padrão de registro. Assim, se houver necessidade de algum log específico, essa configuração deverá ser feita novamente através da ferramenta acima (/system).

Além disso, adotamos como padrão a biblioteca slf4j, que possui uma melhor performance no registro de log. Assim, será necessário atualizar a importação de dependência e alterar o nome da classe em todas as customizações, caso contrário a aplicação não irá iniciar.

Segue um exemplo do padrão anterior, **log4j**:

```
#Importação da dependência  
import org.apache.log4j.Logger;  
...  
#Definição classe  
private Logger logger = Logger.getLogger(NomeClasse.class);
```

E agora, um exemplo no padrão novo, usando **slf4j2**:

```
#Importação da dependência  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
...  
#Definição classe  
private Logger logger = LoggerFactory.getLogger(NomeClasse.class);
```

Utilizar a biblioteca **Slf4j** na versão **1.7.36** (slf4j-api.jar) no ambiente de desenvolvimento das customizações.

Comentários PSP

- As mudanças de dependências e nomes das classes devem ser feitas antes da atualização.
- Alterar nos robôs, integrações e classes
- Pode ser feito com find e replace no projeto.

Item que impacta na inicialização do ambiente

Alteração do tipo do campo código de usuário



Ao iniciar o ambiente será executado o liquibase para alterar o tipo de dados de todas as colunas relacionadas ao usuário de todas as tabelas do BD. Isso pode demorar para ser executado.

No BD MySQL e SQLServer, o tipo do campo foi alterado para bigint e no Oracle para Number(19,0).

Já na programação em Java, o tipo do campo foi alterado de short para long. Devido a essa alteração, é necessário rever todas as customizações (robôs, integrações, aplicações externas, portais, webhooks etc.) que manipulam todos os campos relacionados ao usuário. Caso não sejam ajustadas, o código poderá ser convertido errado e o ambiente não irá iniciar.

Segue abaixo a relação dos nomes dos campos que foram alterados:

- ABERTO_POR
- COD_LIDER
- COD_REMETENTE
- COD_SUBSTITUTO
- COD_USUARIO
- COD_USUARIO_ETAPA
- COD_USUARIO_PROPRIETARIO
- CODIGO_ABERTO_POR
- CODIGO_PARADO_EM
- MULT_CODIGO
- USER_ID
- USUARIO
- USUARIOPROPRIETARIO

Comentários PSP

- Pode ser feito com find e replace no projeto
- Alterar Short e Integer para Long e short e int para long
- Os campos listados ao lado podem variar de acordo com o projeto de cada cliente

Item que impacta na inicialização do ambiente

Alteração do tamanho dos campos nos modelos

Deve ser feita uma análise dos modelos no ambiente do cliente, para verificar quais campos recebem o código do usuário, pois será necessário alterar o tamanho dos mesmos para 9.

Todos os campos devem ser alterados, incluindo os inativos, invisíveis, ocultos ou de modelos suspensos.

Vale lembrar que nem sempre os códigos, versões e campos dos modelos são iguais para os ambientes de homologação e produção. Por este motivo, a criação das query's deve ser feita separadamente.

Após a aplicação de todas as query's será necessário gerar tabela para os modelos. Para essa ação podem ser usados os atalhos disponíveis no módulo de Studio > Gerar Tabelas:

- Para todos os processos ativos
- Para os processos de maior versão

Comentários PSP

- Tomar cuidado em relação à quantidade de modelos no ambiente do cliente, uma grande quantidade pode acabar gerando lentidão na utilização dos atalhos de geração de tabelas.

Mudança na utilização de documentos estáticos

A partir de agora, a utilização de documentos estáticos deve ser feita via ECM. Logo, você deve criar um template, no módulo Documentos, onde serão inseridos os arquivos usados. Para montar a URL de download, utilizada no arquivo JS, você precisa pegar o ID do documento que pode ser encontrado ao clicar com o botão direito em cima do mesmo, na opção Visualizar > Documento. Segue base de URL para download de arquivos estáticos:

- <https://hostAplicacao/bpm/view?dl=true&documentId=idDocumento>

Você também pode utilizar a forma de compartilhamento do documento para gerar essa URL, disponível em Documentos, botão direito em cima do arquivo, na opção Visualizar > Versões, botão direito em cima da versão, Compartilhar > Criar Compartilhamento.

Caso não seja possível fazer dessa forma, você pode utilizar um controller para mapear os arquivos estáticos. Para fazer dessa maneira, você deverá inserir os arquivos no servidor no caminho abaixo:

- </opt/lecom/custom/static-files>

Utilizando o comando `bpm -u update` os arquivos serão sincronizados na pasta do caminho `/opt/lecom/storage/static-files`, onde o controller fará o acesso para pegar o arquivo estático. Segue um exemplo de URL para chamada do controller:

- <https://hostAplicacao/bpm/upload/cadastros/templates/arquivo.pdf>

Um exemplo de controller para esse mapeamento pode ser encontrado no Teams em *Time Operações > Arquivos > Clientes > Pylon > Paschoalotto > Migração 5.50 1.05*.

Atualização da versão do Spring Boot para 2.6.6

Atualização dos componentes



Devido à atualização do Spring Boot, foi necessário atualizar os componentes `integracao.jar`, `RoboWF.jar` e `WfMail.jar`, que são utilizados nas customizações, caso contrário a aplicação não irá iniciar. Esses componentes estão disponíveis através do Nexus (repositório de componentes e bibliotecas), através dos links abaixo:

<https://repo-store.lecom.com.br/#browse/browse:lecom-releases:br%2Fcom%2Flecom%2Fintegracao%2F1.6.0-RELEASE>

<https://repo-store.lecom.com.br/#browse/browse:lecom-releases:br%2Fcom%2Flecom%2FRoboWF%2F1.4.0-RELEASE>

<https://repo-store.lecom.com.br/#browse/browse:lecom-releases:br%2Fcom%2Flecom%2FWFMail%2F1.5.0-RELEASE>

Item que impacta na inicialização do ambiente

Nova versão do driver do MySql

Como foi atualizada a versão do driver do MySql, para obter todos os benefícios dela, recomendamos a atualização das conexões dos BD's auxiliares e/ou externos. Para fazer isso, utilize o ConfigBD, através da opção "J Configuração Livre".

Segue o padrão anterior do driver utilizado:

com.mysql.jdbc.Driver

E agora o padrão novo:

com.mysql.cj.jdbc.Driver

Comentários PSP

- Todos os properties poderão ser atualizados logo após subir a nova versão, com exceção o do Workflow que será atualizado pela equipe de produto

DevOps (apenas Lecom)

Atualização das rotas

Com as alterações do Spring-boot, foi necessário ajustar as chamadas do BPM para utilizar as novas rotas de definição dos templates de customização. Para cada projeto existente que utiliza o DevOps, deverá ser ajustado o import no arquivo pom.xml

Segue o padrão anterior feito no arquivo pom.xml:

```
<parent>
  <groupId>br.com.lecom</groupId>
  <artifactId>integration-parent</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <relativePath/>
</parent>
```

E agora o padrão novo:

```
<parent>
  <groupId>br.com.lecom</groupId>
  <artifactId>integration-parent</artifactId>
  <version>1.0.0-RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

Sequência de atividades para atualização



Pré-requisito: No ambiente do cliente, realizar a alteração de log das customizações **antes** de iniciar a migração.

Depois disso feito, prosseguir com a atualização da release no ambiente do cliente.

Links úteis

Help Center

<https://lecomsa.atlassian.net/wiki/spaces/PLV5/pages/1837334529/Release+vers+o+-+5.50+RTM+1.05>

Inside

<http://inside.lecom.com.br/bpmecm/5-50-rtm-1-05/>

Obrigado!

Produto e Tecnologia



O SENTIDO DA TRANSFORMAÇÃO



www.lecom.com.br