


Construindo portais com Plone

Módulo 11
Pacote Policy

A thick, solid red horizontal bar with rounded ends, positioned below the text 'Módulo 11' and 'Pacote Policy'.

Organização do Projeto

Uma maneira bastante usada para organizar logicamente um projeto Plone é dividir o seu projeto em três pacotes:

`nomedoprojeto.theme`: Onde fazemos customizações visuais.

`nomedoprojeto.content`: Onde criaremos novos tipos de conteúdo.

`nomedoprojeto.policy`: Customizações no site Plone.

O que faremos com nosso pacote policy?

- Gestão de dependências da aplicação
- Definir estrutura inicial de pastas
- Customizar mapeamento de papéis e permissões
- Adição de grupo
- Adição de usuários
- Customizar as actions
- Customizar portlets

Exercício 1: Criando o pacote

- Utilize o esqueleto plone do ZopeSkel para criar o pacote treinamento.policy.
- `../bin/zopeskel plone treinamento.policy`
 - Register Profile: True
 - Todas as outras perguntas aceite o padrão ([Enter])
- Adicione ao pacote (eggs e develop de buildout.cfg)
- Execute o buildout
- Reinicie a instância e verifique se o pacote está disponível para ser instalado.

Exercício 2: Mudando o título do pacote

- Abra o arquivo `treinamento.policy/treinamento/policy/configure.zcml`
- `title:`
 - `treinamento.policy`
 - Projeto Alfa - Policy
- `description:`
 - Installs the `treinamento.policy` package
 - Transforma um site Plone em um Projeto Alfa
- Reinicie a instância e verifique as mudanças.

Configurando dependências

- Podemos utilizar o pacote policy para instalar as dependências do nosso projeto, inclusive os outros pacotes desenvolvidos.
- Para fazer isso, utiliza-se dois arquivos:
 - `treinamento.policy/setup.py`: Faz com que a dependência esteja disponível para ser instalada.
 - `profiles/default/metadata.xml`: Faz com que a dependência seja instalada no momento da instalação do pacote.

Exercício 3: Configurando dependências

- No arquivo `treinamento.policy/setup.py`, adicione os pacotes `treinamento.theme` e `treinamento.content` à lista `install_requires`.
- Em `metadata.xml`, abaixo de `<version>`, adicione o seguinte trecho:

```
<dependencies>
```

```
  <dependency>profile-treinamento.theme:default</dependency>
```

```
  <dependency>profile-treinamento.content:default</dependency>
```

```
</dependencies>
```

Instale o pacote `treinamento.policy` em um novo site e veja resultado.

GenericSetup

- É usado para aplicar customizações de pacotes na configuração do site.
- Utiliza XML para importar e exportar configurações do site no Plone.
- Várias configurações podem ser feitas usando GS, por ex: workflow, portlets, actions, skins, etc.
- Não é necessário editar “à mão” arquivos XML, você pode realizar as configurações TTW e exportá-las usando a aba Export em portal_setup.

GenericSetup (2)

- Mostrar configure.zcml, onde é registrado um perfil para o GenericSetup.
- Mostrar a pasta profiles/default.

GenericSetup (3) > Export

- Mostrar a exportação de um XML do GenericSetup.

Exercício 4: Modificando o workflow padrão

- Vamos colocar o intranet_workflow como workflow padrão do site.
- Primeiro, faça essa modificação na ZMI, em portal_workflow (procure pelo campo Default).
- Agora, na aba Export do portal_setup, exporte o Workflow Tool.
- Pegue o arquivo workflows.xml e coloque em seu profiles/default.
- Remova tudo que for desnecessário.

GenericSetup Steps

- As configurações feitas por um profile do GenericSetup são divididas em passos (ou steps)
- Exemplos:
 - Workflow (workflows.xml)
 - Registro de tipos (types.xml)
- Podemos criar nossos próprios passos para serem executados no momento da instalação do nosso pacote.

Exercício 5: Setuptools

- Exercício 5

plone.api

- É uma api simples e elegante para realizar tarefas comuns de desenvolvimento.
- Documentação:
 - api.plone.org
- Para instalá-la:
 - Adicione **plone.api** no eggs do seu buildout.
 - Rode o buildout (-c develop.cfg).

Exercício 6: plone.api

- Explorar a api usando @@ipdb.
- Utilize o autocomplete [Tab] para ver os atributos/métodos de cada objeto.
- Alguns para serem testados:
 - `api.user.get_current()`
 - `api.user.get_roles()`
 - `api.user.get_users()`
 - `api.group.get_groups()`
 - `api.content.get_state(context)`

Exercício 7: Removendo objetos

- Exercício 7

Exercício 8: Adicionando conteúdo

- Exercício 8

Exercício 9: Criando um grupo

- Exercício 9

Exercício 10: Criando usuários

- Exercício 10

Perguntas?

