Dependências de projetos Python: requirements.txt

Jan 6, 2018

Depois de uma conversa longa outro dia sobre dependências e reproducibilidae de ambientes, decidi fazer essa colinha pra falar de um arquivo muito frequente nos projetos Python (http://python.org/) que vejo por aí: o `requirements.txt`.

Um passo comum da configuração de ambientes de desenvolvimento Python é fazer a instalação de dependências. Muitas vezes esse passo é executado da seguinte forma:

```
$ pip install -r requirements.txt
```

Mas o que é de fato esse arquivo? Segundo a documentação do pip sobre arquivos requirement (https://pip.pypa.io/en/stable/user_guide/#requirements-files):

> "Requirements files" are files containing a list of items to be installed using pip install

Ou seja, esse arquivo nada mais é do que um arquivo de texto, contendo uma lista de itens/pacotes para serem instalados durante o `pip install`.

Vamos a um exemplo

Considerando esse `requirements.txt` aqui:

```
1    -e git+https://github.com/jtemporal/caipyra.git@master#egg=caipyra (https://github.com/jtemporal/caipyra.git@master#egg=caipyra)
3    seaborn==0.8.1
2    pandas>=0.18.1
4    serenata-toolbox
```

Temos várias informações importantes, vamos repassá-las linha a linha:

1. **-e git+https://github.com/jtemporal/caipyra.git@master#egg=caipyra (https://github.com/jtemporal/caipyra.git@master#egg=caipyra)**: Dessa forma conseguimos instalar pacotes Python que estejam disponíveis no GitHub mas não no PyPI. Essa é uma dica muito legal quando por exemplo, você precisa da versão em desenvolvimento de um biblioteca ou quando você prefere usar um fork no lugar da versão tradicional do pacote.

2. **seaborn==0.8.1**: No caso do seaborn, estamos instalando a versão `0.8.1`. Fixar a versão dessa forma é interessante pois garante que o seu projeto vai sempre estar funcionando já que mudanças nos pacotes são indicadas pela alteração no número da versão.

3. **pandas>=0.18.1**: Da mesma forma que o sinal de `==` define uma versão específica a ser instalada, quando usamos o sinal de `>=` nessa lista estamos dizendo que queremos instalar qualquer versão da biblioteca, nesse caso o pandas, seja ela a versão `0.18.1` ou uma mais recente. Interessante nesse caso é notar que você pode definir um intervalo de versões, por exemplo, `pandas>=0.15.0,<=0.18.1`.

4. **serenata-toolbox**: Quando não especificamos uma versão, o pip sempre tentará instalar a versão mais recente do pacote especificado.

Criando seu requirements.txt

Um jeito simples de criar o seu próprio arquivo de depências é usando o comando `pip freeze` da seguinte forma:

```
$ pip freeze > requirements.txt
```

O comando `pip freeze` lista no terminal os pacotes Python instalados no seu ambiente já no formato que o pip install consegue entender, ao associá-lo com o operardor de redireção `>` você consegue escrever a mesma lista que aparece no terminal dentro do arquivo de texto.

Massa né? Agora é só criar arquivos de dependências para todos projetos 😜

Links

- Caipyra (https://github.com/jtemporal/caipyra)
- pandas (https://pandas.pydata.org/)
- seaborn (https://seaborn.pydata.org/)
- serenata-toolbox (https://github.com/datasciencebr/serenata-toolbox)

- Artigo em inglês sobre operadores de controle e redireção (https://unix.stackexchange.com/questions/159513/what-are-the-shells-control-and-redirection-operators)

Agradecimentos

Obrigada aos amigos que participaram da conversa sobre dependências e reproducibilidade de ambientes <3

« Escolhendo um tema Jekyll (https://jtemporal.com/temas-jekyll/)

Forçando o rebuild de sites Jekyll hospedados no GitHub » (https://jtemporal.com/force-rebuild-jekyll/)

Recent articles

Copiando arquivos para dentro do container (https://jtemporal.com/copiando-arquivos-para-dentro-do-container/) 03 Apr 2019

Brincando com a listagem de containers Docker (https://jtemporal.com/brincando-com-a-listagem-de-containers-docker/) 16 Mar 2019

Diferenciando json.loads de json.load e uma pitada de BigQuery (https://jtemporal.com/diferenciando-json-loads-de-json-load-e-uma-pitade-de-bigquery/) 15 Mar 2019

Built by webjeda (http://webjeda.com)

(http://twitter.com/jesstemporal) (http://github.com/jtemporal)

(mailto:jessicatemporal@gmail.com)