



Para saber mais: Tipos de versionamento

No Django Rest, existem algumas formas de versionar uma API. Vejamos a seguir alguns exemplos:

QueryParameterVersioning

Transfere a versão através do parâmetro **version**

Exemplo:

```
http://exemplo.com/alunos/?version=1
```

```
http://exemplo.com/alunos/?version=2
```

COPIAR CÓDIGO

HostNameVersioning

A versão é definida pelo nome de domínio

Exemplo:

```
http://v1.exemplo.com/alunos/
```

```
http://v2.exemplo.com/alunos/
```

[COPIAR CÓDIGO](#)

NamespaceVersioning

A versão é fornecida através do namespace da url

Exemplo:

```
urlpatterns = [  
    url(r'^v1/alunos/', include(alunos.  
        url(r'^v2/alunos/', include(alunos.  
    ]
```

[COPIAR CÓDIGO](#)

URLPathVersioning

Adiciona a versão no endereço do recurso de uma variável (o caminho é verificado através do parâmetro VERSION_PARAM)

Exemplo:

```
urlpatterns = [  
    url(r'^(?P<version>(v1|v2))/alunos/  
        name='alunos-lista'  
    )  
]
```

COPIAR CÓDIGO

AcceptHeaderVersioning

Transferência do número da versão através do cabeçalho da requisição

Exemplo:

```
GET /alunos/ HTTP/1.1  
Host: exemplo.com  
Accept: application/json; version=1.0
```

[COPIAR CÓDIGO](#)

Não podemos esquecer de incluir o tipo de versionamento na variável de ambiente do

`REST_FRAMEWORK` no arquivo `settings.py`.

Por exemplo:

```
REST_FRAMEWORK = {  
    'DEFAULT_VERSIONING_CLASS': 'rest_f  
}
```

[COPIAR CÓDIGO](#)