

# API DO VERIFICADOR DE CONFORMIDADE

---

## TUTORIAL DE UTILIZAÇÃO

**A API, que retorna o relatório em formato JSON, tem dois `_endpoints_`:**

- 1. O endereço raiz “/”**, onde utilizamos de uma heurística interna para classificar se o arquivo enviado é de fato assinado com algo suportado pelo Verificador (ou seja, certificado ICP-Brasil).
- 2. O endereço “/report”**, que retorna o relatório em si. Focaremos mais nesta parte, assumindo que os artefatos enviados são de cunho específico e provavelmente estarão assinados corretamente.

**Para utilizá-la, seguem exemplos abaixo utilizando a ferramenta cURL**

O primeiro exemplo envia duas assinaturas anexadas de tipos quaisquer e retorna um JSON com ambos os relatórios, sem quaisquer limitação no número das mesmas. (Se necessário, é trivial configurar esse limite.) Os arquivos devem ser enviados pelo POST com o cabeçalho Content-Type sendo `multipart-form-data`

```
curl -L -v -F "signature_files[]=@<sig.pdf>"
-F "signature_files[]=@<sig.p7s>" example.com/report
```

O segundo exemplo envia um par de arquivos que caracterizam uma assinatura destacada. O funcionamento é o mesmo, mas o Verificador não aceita arquivos anexados e destacados misturados na requisição.

```
curl -L -v -F "signature_files[]=@<sig.p7s>"
-F "detached_files[]=@<det_sig.dat>" example.com/report
```

### Sobre a imagem Docker

É possível rodar o Verificador facilmente com o comando abaixo. De fato, não é uma versão simplificada com apenas a API, e contém a interface web também, mas está totalmente funcional após testes rápidos. A partir do momento que obtiverem os arquivos WAR, essa imagem pode ser trivialmente construída e inserida em infraestruturas de testes.

```
docker run -v
/path/to/verifier.war:/usr/local/tomcat/webapps/verifier.war
-it -p 8080:8080 tomcat:8.5.53-jdk8-openjdk-slim
```

**\* É necessário inserir a cadeia de certificação correta ao fazer requisições, visto que a AC Raiz da mesma não está inserida nos navegadores e sistemas operacionais por padrão.**