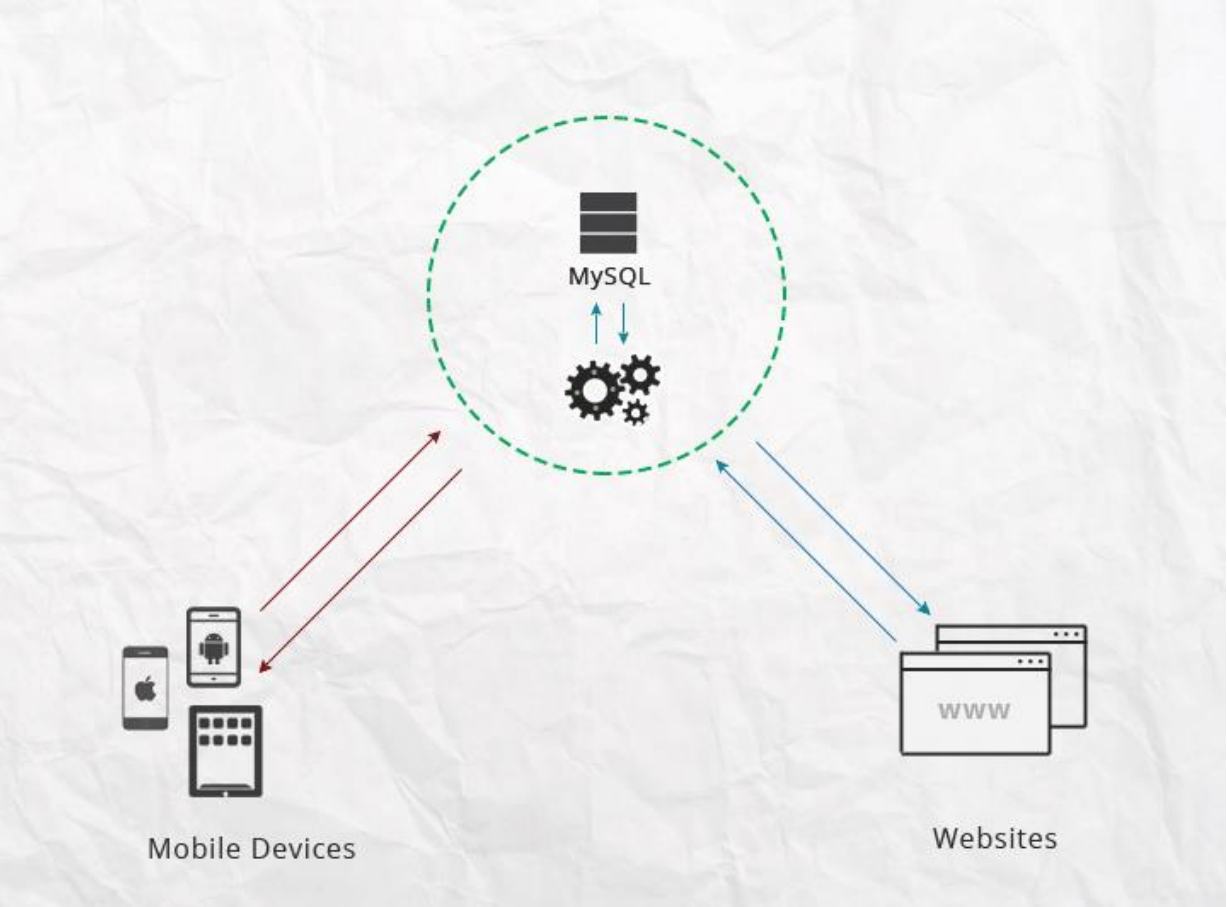
targettrust
treinamento e tecnologia

Webservices e APIs

REST

Abreviação de "REpresentational State Transfer"


MySQL
WWW
Mobile Devices
Websites

REST

REST é um conjunto de princípios que definem como Web Standards como HTTP e URIs devem ser usados.

REST

O que é importante ter em mente é que o principal em um *restful web service* são as URLs do sistema (geralmente são referidas como *restful url's*), e os *resources*. Um *resource* é um recurso, uma entidade, ou seja, é um objeto com informação que será representado por meio de um XML. Em geral, a URL para acessar esse recurso será sempre a mesma, porém caso mudemos o método HTTP (GET, POST, PUT, DELETE) o resultado da requisição será diferente.

REST

Método	exemplo.com/recurso	exemplo.com/recurso/1
GET	lista os recursos	detalhes de um recurso
POST	adiciona um recurso	-
PUT	-	atualiza um recurso
DELETE	-	remove um recurso

REST

200	OK
201	Created
304	Not Modified
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
422	Unprocessable Entity
500	Internal Server Error

REST

Porém saiba que no REST não existe um padrão definido, as operações listadas acima são apenas uma sugestão. Ao desenvolver um *web service* você não é obrigado a implementar todas as operações para seguir um determinado padrão, você só precisa implementar o que é necessário no seu contexto.

REST

Em REST projetar as URLs de endpoint, elas devem ser bem formadas e devem ser facilmente compreensíveis.

Cada URL para um recurso deve ser identificados de forma exclusiva. Se a sua API precisa de uma chave de API de acesso, a chave de API deve ser mantido em cabeçalhos HTTP em vez de incluí-lo na URL.

REST

Há uma enorme discussão sobre o versionamento de a API , se manter versão api na URL ou nos cabeçalhos de solicitação HTTP. Mesmo que ele é recomendado que a versão deve ser incluída nos cabeçalhos de solicitação, eu me sinto confortável para mantê-lo na URL em si, pois é muito conveniente no lado do cliente para migrar de uma versão para outra.

REST

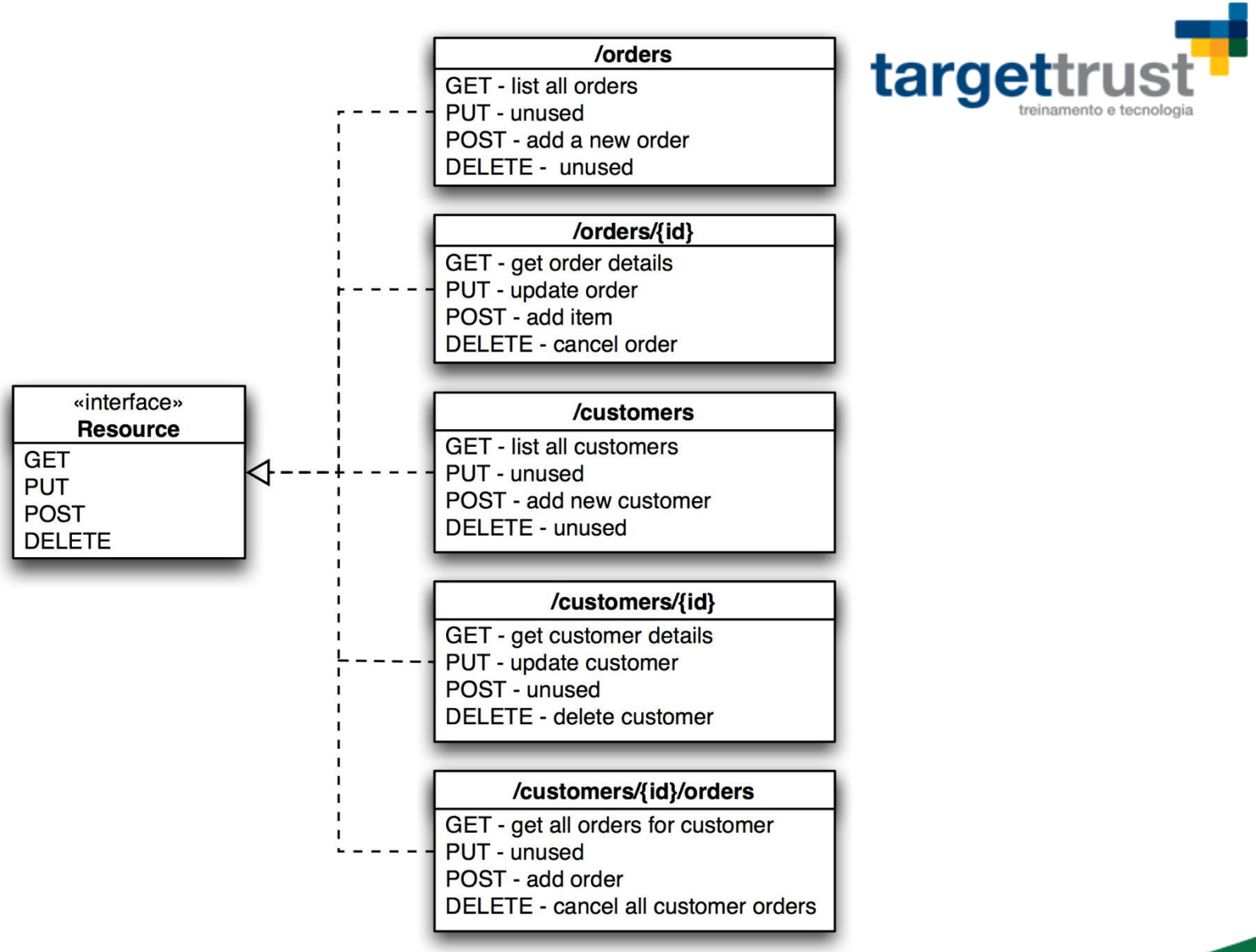
http://abc.com/**v1**/tasks
http://abc.com/**v2**/tasks

REST

O JSON Mime type dev ser **Content-Type: application/json**, para XML **Content-Type: application/xml**. Veja mais Aqui

REST

Por exemplo, eu posso decidir que os recursos não podem ser apagados e não implementar nenhuma operação para o método DELETE. O que deve ser levado em consideração é a convenção das ações. GET deve ser usado para “leitura”, por exemplo listar os detalhes de um recurso. POST deve ser usado para adicionar novos recursos. PUT deve ser utilizado para alterar recursos já existentes. E DELETE para apagar recursos.


«interface»
Resource
GET
PUT
POST
DELETE
/orders
GET - list all orders
PUT - unused
POST - add a new order
DELETE - unused
/orders/{id}
GET - get order details
PUT - update order
POST - add item
DELETE - cancel order
/customers
GET - list all customers
PUT - unused
POST - add new customer
DELETE - unused
/customers/{id}
GET - get customer details
PUT - update customer
POST - unused
DELETE - delete customer
/customers/{id}/orders
GET - get all orders for customer
PUT - unused
POST - add order
DELETE - cancel all customer orders