



Embarcadero Conference

Um único esforço, uma única base de código, múltiplas
plataformas, múltiplos dispositivos



Mário Guedes

Conhecendo o Novo REST Framework

Um novo momento: nuvem



- Estamos vivendo uma nova era: *soluções in cloud*.^[L SEP]
- O usuário final quer a informação onde ele, o usuário, estiver.^[L SEP]
- Os clients agora são Desktop, Web e Mobile! Em Windows, Mac, Linux, Android, iOS, Windows Phone ...^[L SEP]
- Não se esqueça da Internet das Coisas: Tecnologia Vestível, Drone, Automação Residencial, e por aí vai.^[L SEP]
- Temos o maior banco de dados do mundo ao nosso dispor: a grande web e o **BIG DATA** com os seus "Vs"
 - Volume
 - Velocidade
 - Variedade



Web Humana X Web Programável



- Essencialmente existem duas webs. L
SEP
- A “web humana”, feita para ser consumida pelas pessoas. O maior ícone é uma página web.
- *Se preocupa tanto com a apresentação quanto com a informação.* L
SEP
- E a “web programável”, feita para ser consumida por softwares.
- *Tem foco apenas na informação.* L
SEP
- Nada a ver com a Deep Web – Isto é outra coisa.



Momento mão na massa: #EC2014

- Um exemplo extremo é o consumo da **Streaming API** do Twitter. L
SEP
- A Streaming API esta fora do escopo desta apresentação pois envolve uma outra tecnologia: o **WebSocket**, comunicação persistente e bidirecional. L
SEP
- Imagine acompanhar em tempo real os twittes de uma determinada hashtag. L
SEP
- Este é o momento em que você pega o seu celular e twitta algo assim: *"#EC2014 clRed Delphi XE7 – o melhor!"* L
SEP



Protocolo de Transporte e de Comunicação

- **Protocolo:** é uma convenção, ou seja, um conjunto de regras bem estabelecidas, que torna possível a comunicação entre duas partes. L
SEP
- Vamos falar muito de HTTP. E o TCP/IP? Onde fica? L
SEP
- A verdade é que são coisas distintas: L
SEP
 - TCP/IP: É um protocolo de transporte entre dois equipamentos L
SEP
 - HTTP: É um protocolo de comunicação entre dois softwares L
SEP
- Uma mensagem HTTP trafega via TCP/IP entre um equipamento e outro.

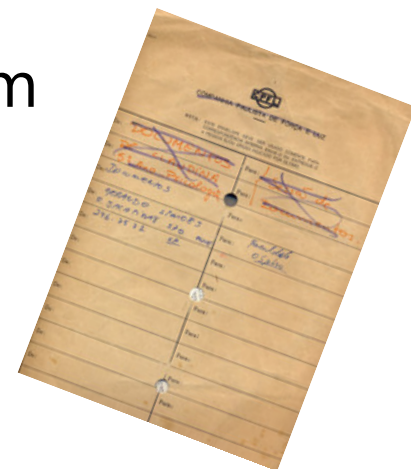


HTTP – Protocolo de Transferência de Hipertexto

- Baseado em texto: simplicidade. [L SEP]
- Orientado à documento: o objetivo primário é o de recuperar um documento de um servidor remoto. [L SEP]
- Fundamentado no modelo de Requisição e Reposta: Síncrono [L SEP]
- Hipertexto é um texto com referência a outros textos. [L SEP]
- Surgiu junto com o HTML, nos anos 90. [L SEP]
- HTML: *Linguagem de marcação de hipertexto*

Pense no HTTP como um envelope de vai e vem

- Uma mensagem HTTP tem um remetente e um destinatário. L
SEP
- E talvez um conteúdo. L
SEP
- Durante o tráfego a mensagem passa por diversos softwares e hardwares, como o proxy da sua empresa. L
SEP
- Então o cabeçalho HTTP possui diversas informações para que os softwares envolvidos tomem as melhores decisões.



Regras gerais do protocolo HTTP

- Uma mensagem HTTP possui três partes bem distintas:

1 – Cabeçalho HTTP

POST /mensagem HTTP/1.1<

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) ...<

Content-Type: application/json<


Content-Length: 22<

2 - Quebra de linha
indicando o fim do
cabeçalho

<
{“texto”: “olá mundo”}

3 – Conteúdo da mensagem,
se houver

Requisição HTTP

- O cabeçalho HTTP de uma requisição possui o seguinte formato: 

1 – Verbo HTTP

2 – Caminho do recurso

3 – Versão do protocolo HTTP

GET /index.html HTTP/1.1
Host: www.exemplo.com.br

4 – Pares de chave e valor com informações relevantes

Resposta HTTP

- Já o cabeçalho de uma resposta HTTP possui o seguinte formato:

1 – Versão do protocolo HTTP

2 – Status de retorno do processamento

3 – Mensagem de retorno do servidor

4 – Pares de chave e valor com informações relevantes

```
HTTP/1.1 200 OK<  
Server: Apache<  
Content-Length: 23<
```

```
<  
<html>Olá mundo!</html>
```

5 – Conteúdo da resposta



Alguns cabeçalhos que se destacam

- **Content-Type:** Indica o tipo de conteúdo que esta sendo trafegado. L SEP
- **Content-Length:** Indica o tamanho em bytes do conteúdo que esta sendo trafegado. L SEP
- **Accept:** Indica o tipo de conteúdo que o cliente aceita. L SEP
- **Authorization:** Contêm as credenciais do usuário solicitante. L SEP
- **User-Agent:** Indica o nome e versão do software cliente. L SEP
- **Server:** Indica o nome e versão do software servidor. L SEP
- **Cache-Control:** Indica se pode ou não armazenar o conteúdo em cache.



Momento mão na massa: recuperando a cotação do dólar de um jeito ruim ☹

- O maior banco de dados do mundo é a web.^[L SEP]
- Podemos programar robôs para rastrear sites, fazer cotações automáticas, e por aí vai.^[L SEP]
- Isto é um web crawler.^[L SEP]
- Vamos estudar um que monitora o site www.dolarhoje.com e recupera a cotação do dólar.^[L SEP]
- Funciona, mas a um custo computacional muito alto e temos que torcer para que não mudem o layout da página.



SOA – Arquitetura Orientada à Serviços

- Abordagem extremamente eficiente para a otimização dos recursos de TI de uma empresa. [L SEP]
- Promove a desvinculação entre as regras de negócio e modelos específicos como:
 - Linguagens de desenvolvimento
 - Sistemas operacionais
 - Sistema de banco de dados e por ai vai. [L SEP]
- Interoperabilidade! [L SEP]
- O SOAP é o maior exemplo desta abordagem. [L SEP]
- NFe, quem nunca né? ☺

Consumidor & Provedor



- **Consumidor:** Estrutura de software que promove a visualização das informações bem como a interação com o usuário final. [SEP]
- **Provedor:** Estrutura de software que provê funcionalidades de domínio de negócios: As regras. [SEP]
- **Comunicação:** Para duas partes distintas se comunicarem é necessário haver um meio e um protocolo suportado por ambas.



ROA – Arquitetura Orientada à Recursos

- ROA é o modo de resolver um problema com um serviço web REST. A abstração em relação ao SOA diminui.^[L SEP]
- Agora estamos colocando o REST em evidência e a forma como ele funciona é fortemente aderente ao HTTP.^[L SEP]
- O conteúdo que trafega entre uma ponta e outra é um *recurso*.^[L SEP]
- Evolvabilidade: capacidade de evoluir e se adaptar.

REST – Transferência de Estado Representacional

- *“Um sistema complexo que funciona é, invariavelmente, considerado como evoluído a partir de um simples que funcionava” - John Gall* [LSEP]
- Grosso modo, “o mesmo gênio que inventou o HTTP inventou o REST”: Tim Berners-Lee [LSEP]
- Obviamente existem diversas outras mentes brilhantes neste processo. [LSEP]
- Um servidor REST não guarda estado, cada nó da cadeia:
 - recebe → modifica → repassa o estado de um recurso.
 - Isso se reflete em ESCALABILIDADE!





Interface uniforme

- A ação a ser executada sobre um recurso é determinado pelo verbo HTTP.^{LSEP}
- Os quatro verbos principais equivalem às operações CRUD.^{LSEP}

	Verbo	Ação
Create	POST	Cria um novo recurso
Read	GET	Recupera um determinado recurso ou lista de recursos
Update	PUT	Atualiza um determinado recurso ou cria se inexistente
Delete	DELETE	Elimina um determinado recurso



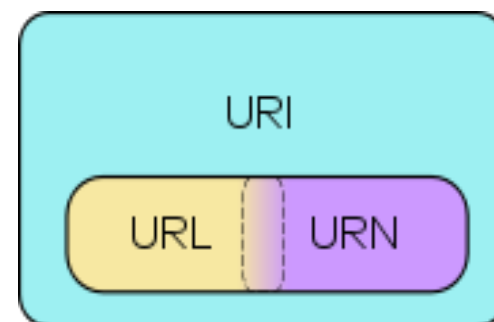
URL? URI?

- Tanto faz. ^LSEP
- Na literatura aparecem os dois, alternadamente. ^LSEP
- A URL, de um modo geral, identifica um recurso disponível em uma rede. ^LSEP

^LSEP

[protocolo]://[usuário]:[senha]@[servidor]:[porta]/[caminho]/[recurso]

URI	Identificador Padrão de Recurso
URL	Localizador Padrão de Recursos
URN	Nome Padrão de Recurso





URL: Recurso a ser trabalho

- A URL identifica o recurso a ser trabalhado. [LSEP]
- Não importa o que será feito com o recurso, a URL será a mesma. [LSEP]
- O Verbo HTTP é que determinará o que será feito. [LSEP]
- As partes de uma URL determinam uma hierarquia em relação ao recurso. [LSEP]
- Para consumir um servidor DataSnap/REST usamos URL por justaposição. [LSEP]

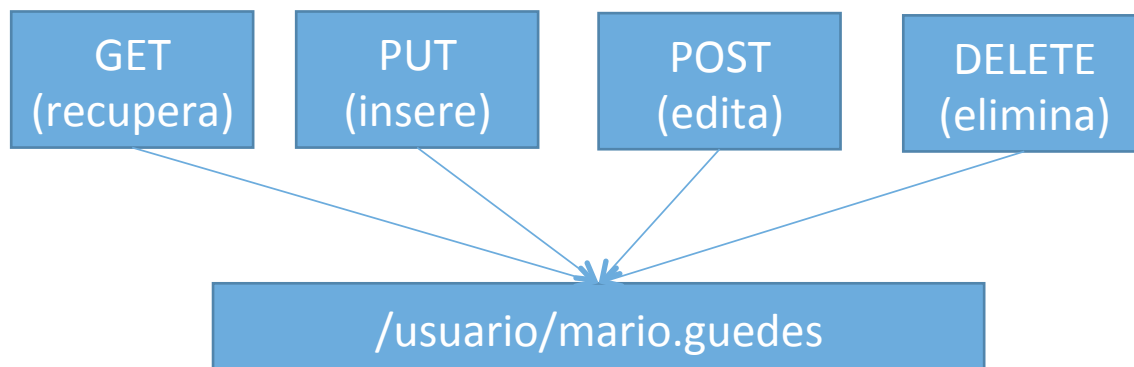


Passagem de parâmetros

- No que concerne ao REST a passagem de parâmetros deve se dar via URL. ^L_{SEP}
- Após a identificação do recurso virá o sinal de ? seguido por pares de <chave>=<valor> ^L_{SEP}
- Os parâmetros são separados pelo & ^L_{SEP}
- Espaço pode ser representado por + ^L_{SEP}
- Caracteres especiais são codificados de acordo com a sua representação UNICODE/UTF8 no formato hexadecimal, após o sinal %: Ç → 199 → C7 → %C7 ^L_{SEP}










Oi?





Principais Status HTTP

- 200 – OK: A solicitação foi processada com sucesso! 
- 201 – CREATED: A solicitação de criação de um recurso (POST) foi executada com sucesso! 
- 202 – ACCEPTED: A solicitação de execução de uma tarefa assíncrona foi enfileirada com sucesso! 
- 304 – NOT MODIFIED: O conteúdo armazenado em cache pela aplicação cliente é válida! 
- 400 – BAD REQUEST: A solicitação possui algum erro por parte da aplicação cliente. 
- 401 – UNAUTHORIZED: O usuário não pode executar a operação desejada. 
- 404 – NOT FOUND: Clássico! O recurso solicitado não existe. 
- 500 – INTERNAL SERVER ERROR: Ocorreu um erro na aplicação servidora.

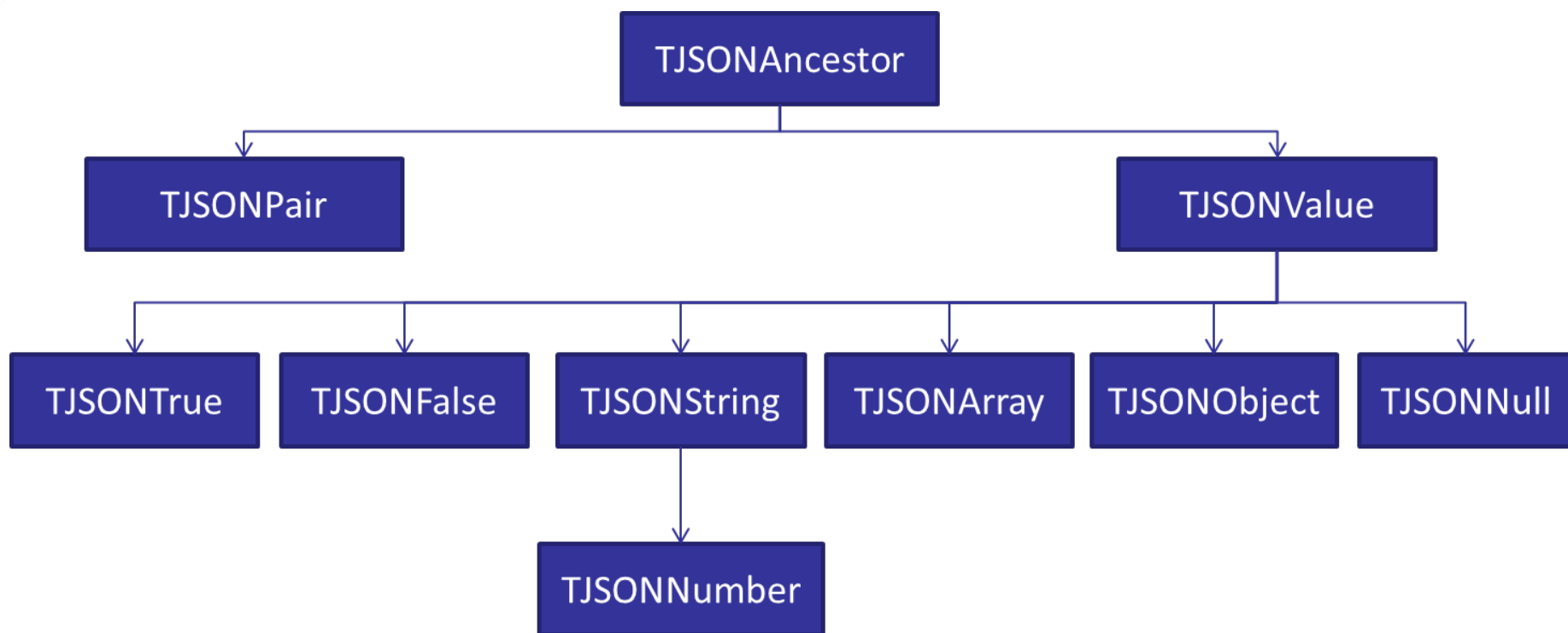


JSON – Notação de Objeto JavaScript

- Um recurso deve ser representado de alguma forma. L
S
E
P
- Mas como podemos representar um vendedor ou uma empresa?
 - Precisamos de uma estrutura complexa. L
S
E
P
- Poderia ser XML, INI, CSV, YAML... Mas a indústria adotou o JSON. L
S
E
P
- Legibilidade! L
S
E
P
- Maior compactação resulta em maior vazão. L
S
E
P
- Aderência ao JavaScript: A linguagem frontend web.

JSON é simples!

- A unit System.JSON possui várias classes para lidar com o JSON no Delphi.





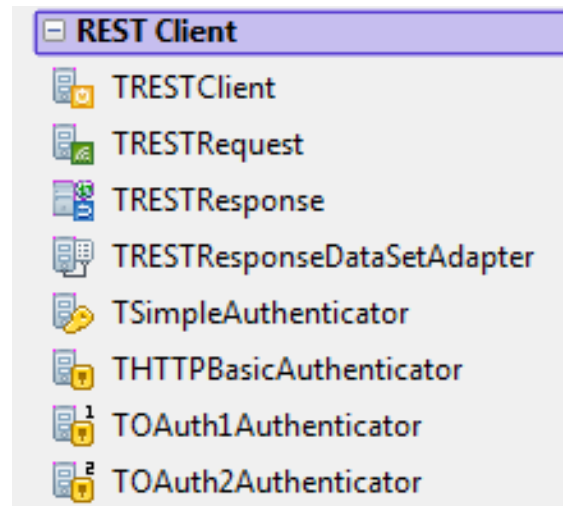
Momento mão na massa: recuperando a cotação do dólar de um jeito bacana 😊

- O ideal é sempre tentarmos consumir um servidor REST para recuperar informações. L
SEP
- Cotações de moedas, previsão do tempo, resultados de jogos ... L
SEP
- Atenção: Fazer um web crawler para alguns sites podem ferir os termos de uso deste. L
SEP
- Vamos rever o nosso sistema de cotação do dólar. L
SEP
- Utilizaremos um serviço web chamado "Cotação API":
<http://api.promasters.net.br/cotacao/>



Caramba! Tenho que fazer tudo isso para consumir um serviço REST?

- Resposta curta: Não! ^L_{SEP}
- Resposta longa: O Delphi XE7 traz consigo um conjunto de componentes que encapsula todas as dificuldades que discutimos até aqui. ^L_{SEP}
- Ufa! ^L_{SEP}
- Menos código na hora de ^L_{SEP} consumir um servidor REST! ^L_{SEP}
- O namespace **REST.*** possui várias facilidades para lidar com toda esta complexidade que falamos até aqui.





Paleta REST Client

- **TRESTClient:** Responsável pela configuração de acesso à um serviço RESTful. L
SEP
- **TRESTRequest:** Responsável pela requisição de um recurso específico. L
SEP
- **TRESTResponse:** Responsável pela decodificação da resposta. L
SEP
- HTTPS: <http://indy.fulgan.com/SSL/>



REST API

- Grandes serviços web possuem uma REST API. L
SEP
- É um conjunto de URLs e representações JSON bem documentadas. L
SEP
- Facebook, Twitter, Produtos Google, LinkedIn, Spotify, Netflix, Instagram, Amazon, GitHub, e provavelmente o seu serviço web preferido possuiu APIs REST. L
SEP
- Deve-se ler a documentação e ficar atento aos termos de uso. Em geral há uma área no site para desenvolvedores. L
SEP
- E o seu sistema pode entrar neste time: DataSnap/REST.



Momento mão na massa: Recuperando os aniversários dos seus amigos do Facebook!

- Vamos usar os componentes da paleta REST Client para listarmos os nosso amigos do Facebook. L SEP
- Agora que já entendemos como funciona os conceitos de HTTP e REST fica mais simples de entender a “mágica”. L SEP
- Para isso temos que usar a API REST do Facebook. L SEP
- <https://developers.facebook.com/>



Dá tempo para perguntas?

- Muito obrigado pelo privilégio da sua audiência! 😊 L
SEP
- Os exemplos poderão ser encontrados em: L
SEP
https://github.com/jmarioquedes/EC2014_REST.git



Referências

- http://pt.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- Livro: REST: Construa API's inteligentes de maneira simples
<http://www.casadocodigo.com.br/products/livro-rest>
Alexandre Saudate
Veja se este código de desconto está válido: PARCEIRO-4tfc3687
- Livro: RESTful Serviços Web
<http://www.altabooks.com.br/restful-servicos-web.html>
Leonard Richardson e Sam Ruby
- Como um verdadeiro sistema REST funciona: arquitetura e desempenho na Abril
<http://www.infoq.com/br/presentations/rest-arquitetura-abril>
Luis Cipriani

HTTP/1.1 200 OK

Content-Length: 544

```
{  
  "nome"      : "Mário Guedes" ,  
  "e-mail"    : "mario.guedes@arrayof.com.br" ,  
  "blog"      : "http://eugostododelphi.blogspot.com.br" ,  
  "perfis"   : [  
    {"twitter" : "eugostododelphi"} ,  
    {"facebook" : "eugostododelphi"} ,  
    {"linkedin" : "jmarioguedes"} ,  
    {"slideshare" : "jmarioguedes"} ,  
    {"github" : "jmarioguedes"} ,  
    {"prezi" : "jmarioguedes"}  
  ]  
}
```