



Getting Things Done with REST

Ian Robinson

<http://ianSrobinson.com>

@ianSrobinson

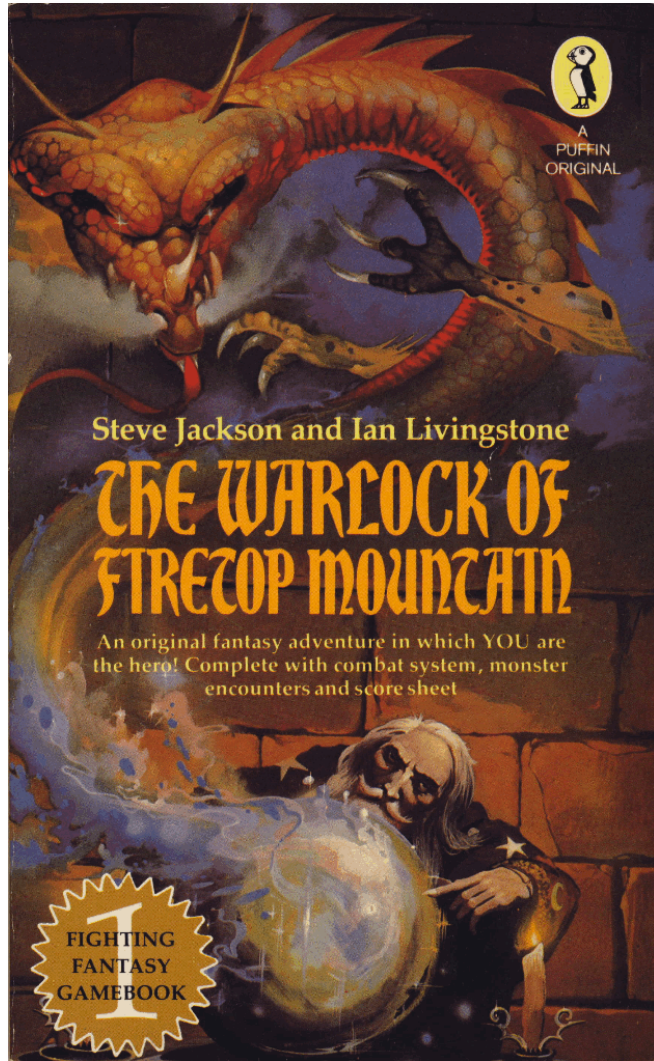
ianSrobinson@gmail.com

ThoughtWorks®

Getting Things Done



Pick your path



79-80

hinges. Listening at the door, you hear strange mutterings and the clatter of what could be pots and pans. Whatever is in there, there are several of them. Do you want to go through the door (turn to 159) or turn back (turn to 237)?

79

The passageway ends in front of you in a dead end. If you wish to search for secret passageways, turn to 137. If not, return to the crossroads at 267

80

The key fits the lock and opens the door. You find yourself in a large boathouse. Various boats, in different stages of construction, are lying around. Apart from the door behind you, there is another in the north wall. As you enter, the Skeletons stop their work and crane their bony necks around to look at you. They pick up planks of wood and hammers and advance towards you. There are five of them. Do you.

- | | |
|--|-------------|
| Smile nervously and back out of the door into the passage? | Turn to 129 |
| Tell them you've come about buying a boat? | Turn to 123 |
| Tell them you're their new boss and order them back to work? | Turn to 195 |
| Draw your sword and prepare for battle? | Turn to 140 |

195-197

flies through the air directly at him, stops centimetres from his chest and falls to the floor. He looks up and smiles at you with an evil, gloating smile. What can you do:

- | | |
|--|-------------|
| Draw your sword and advance? | Turn to 142 |
| Try something else from your backpack? | Turn to 105 |

195

This is a rather unlikely story, considering that they see very few humans around. Nevertheless, Skeletons are pretty dim - you knew this and that's why you tried the story. Roll one die. If you roll a 1 or 2, they don't believe you and keep on advancing. Turn to 140.

A 3 or 4 means that they aren't sure, and send two of their number off through the north door whilst the rest hold you at bay with their weapons. Turn to 164

A roll of 5 or 6 means they've believed you and they all get back to work! Turn to 9. Add 2 LUCK points

196

You search the room. Try as you may you cannot find the secret switch to open the door in the bookshelf - the old man must have locked it from the inside. You do find 5 Gold Pieces in a drawer in the table. You decide to return to the junction to the south. Turn to 280

197

At the top of the stairs the passage turns sharply to

Warlock of Firetop Mountain Protocol

- Go north
- Defeat goblin
- Take key
- Unlock door
- Go east
- Solve riddle

Fighting Fantasy Uniform Interface

- Numbered prose paragraphs
- Multiple choices keyed to numbered paragraphs

Hypermedia

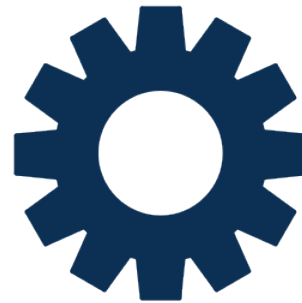
As
the

Engine

of

Application

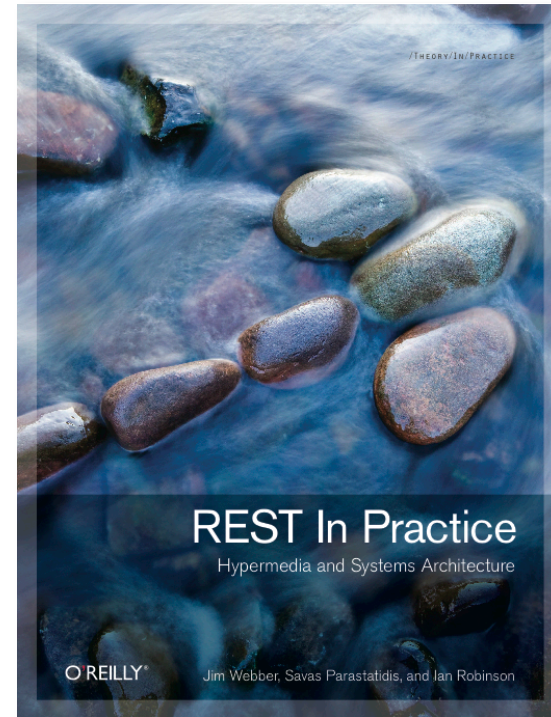
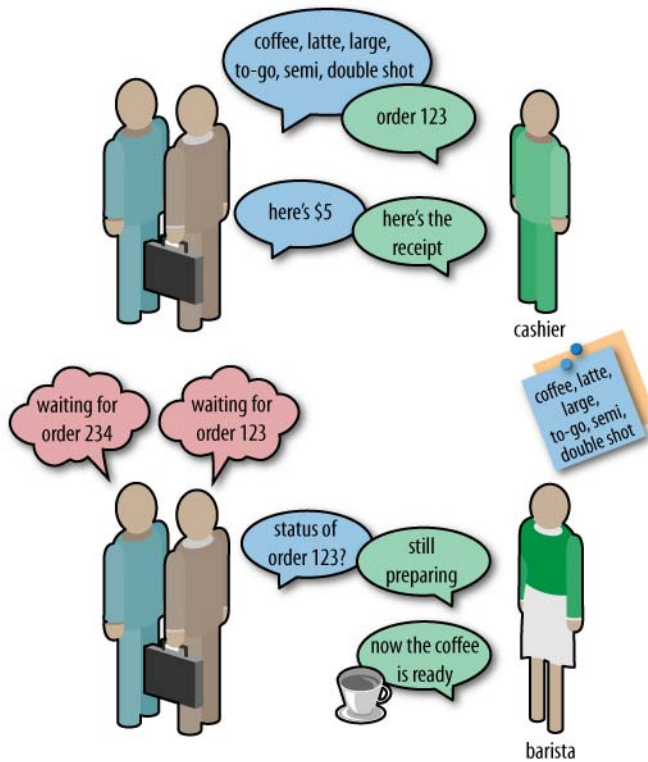
State



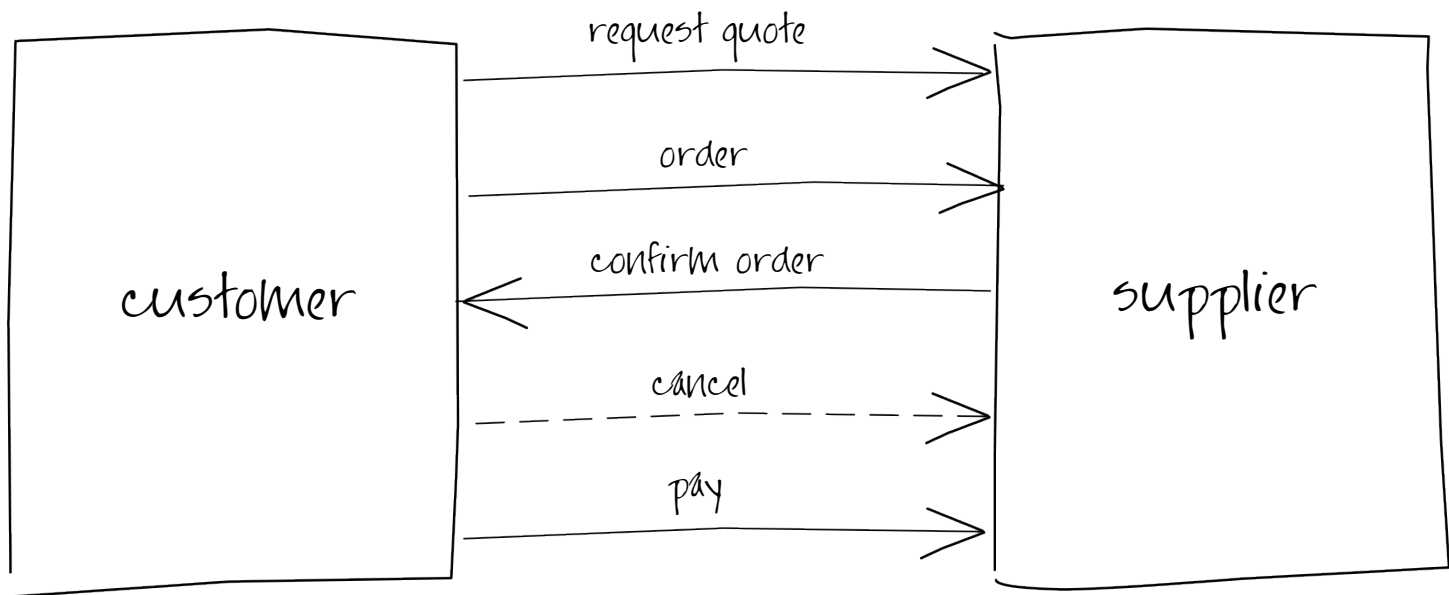
A Procurement Application



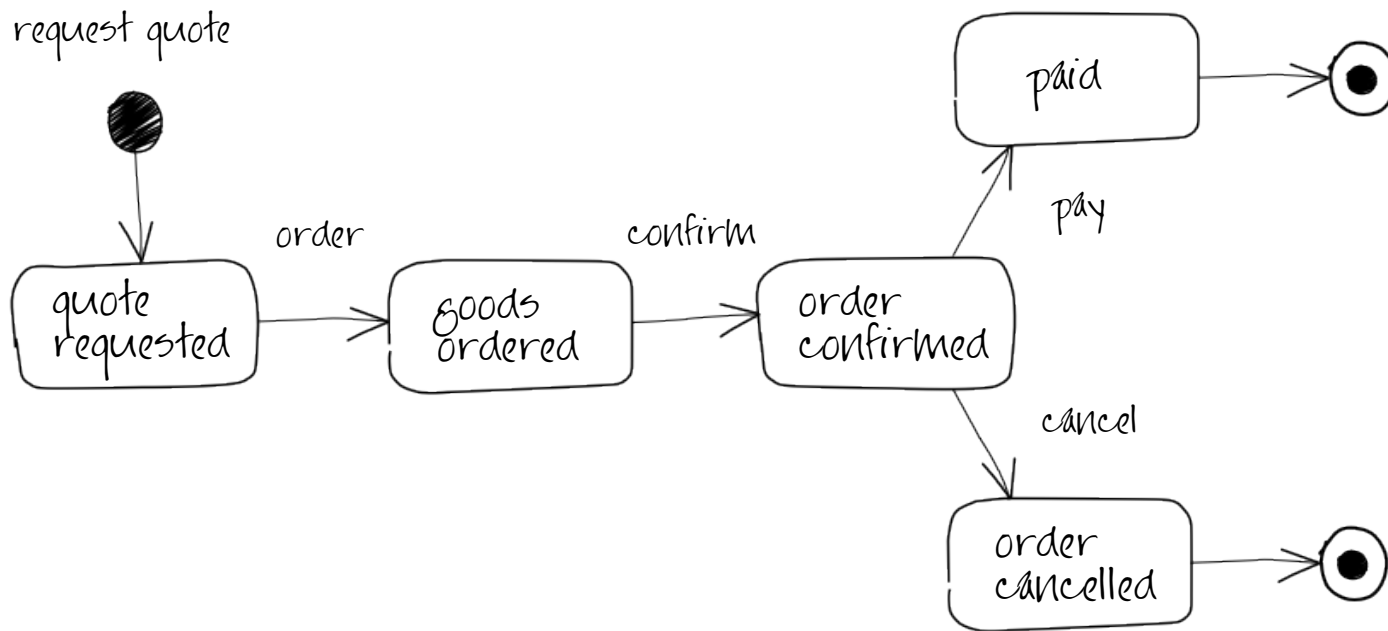
Restbucks



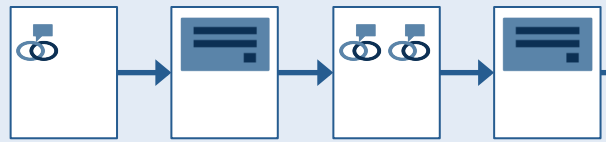
Procurement



Procurement application state transitions



Three capabilities



Quote



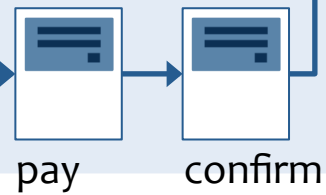
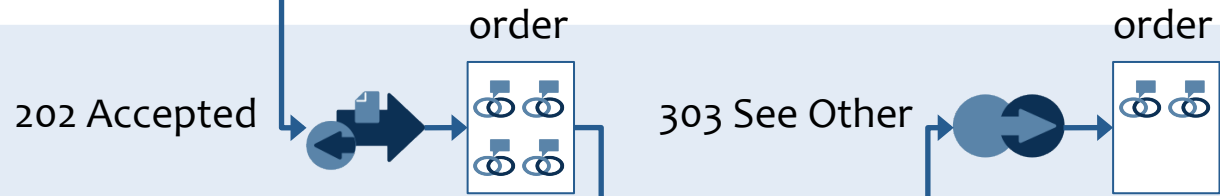
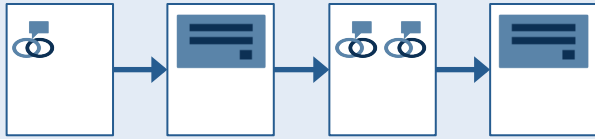
Order



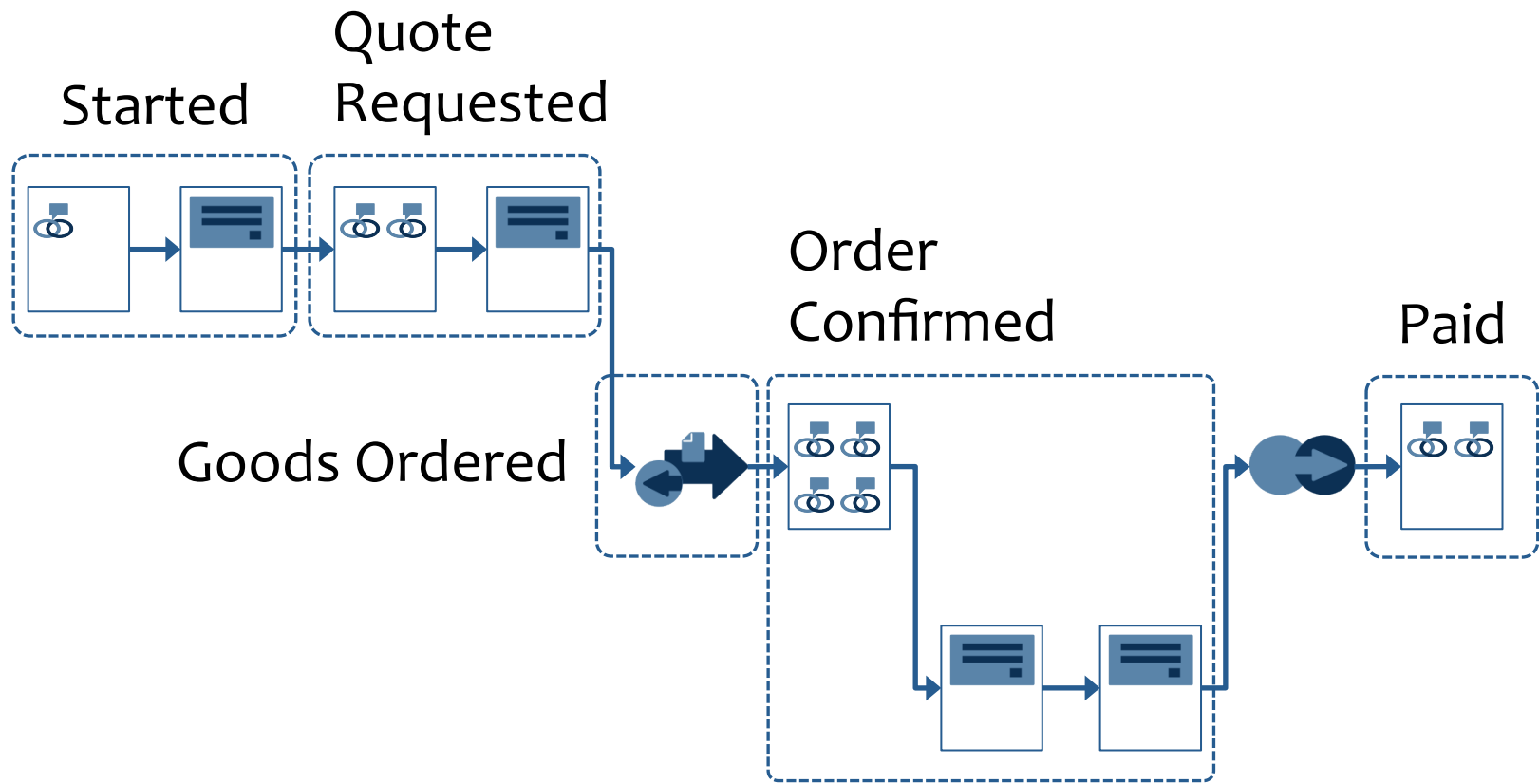
Pay

Many resources

home rfq quote order-form



Client view of application state



‘Designing a RESTful Domain Application Protocol’ in
REST: From Research to Practice (Springer)

Server-side Development



Quote resource (outline)

```
[ServiceContract]
[UriTemplate("quote", "{id}")]
public class Quote
{
    private readonly UriFactory uriFactory;
    private readonly IQotationEngine quoteEngine;

    public Quote(UriFactory uriFactory, IQotationEngine quoteEngine)
    {
        this.uriFactory = uriFactory;
        this.quoteEngine = quoteEngine;
    }

    [WebGet]
    public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage
        response)
    {
        //Get quotation from quotation engine
        //Add HTTP headers to response
        //Return entity body
    }
}
```

Easy to test

```
[Test]
public void ShouldReturn404NotFoundWhenGettingQuoteThatDoesNotExist()
{
    var id = Guid.Empty;

    var quoteEngine = MockRepository.GenerateStub<IQuotationEngine>();
    quoteEngine.Stub(e => e.GetQuote(id))
        .Throw(new KeyNotFoundException());

    var response = new HttpResponseMessage();

    var quoteResource = new Quote(DefaultUriFactory.Instance, quoteEngine);
    quoteResource.Get(
        id.ToString("N"),
        new HttpRequestMessage(),
        response);

    Assert.AreEqual(HttpStatusCode.NotFound, response.StatusCode);
}
```

Return 404 Not Found when quotation doesn't exist

```
public class Quote
{
    private readonly IQotationEngine quoteEngine;

    ...

    [HttpGet]
    public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage
        response)
    {
        Quotation quote;
        try
        {
            quote = quoteEngine.GetQuote(new Guid(id));
        }
        catch (KeyNotFoundException)
        {
            response.StatusCode = HttpStatusCode.NotFound;
            return null;
        }

        ...
    }
}
```


Test caching headers

```
[Test]
public void ResponseShouldExpire7DaysFromDateTimeQuoteWasCreated()
{
    var id = Guid.Empty;

    DateTimeOffset createdDateTime = DateTime.Now;
    var expiryDateTime = createdDateTime.AddDays(7.00) .UtcDateTime;

    var quoteEngine = MockRepository.GenerateStub<IQuotationEngine>();
    quoteEngine.Stub(e => e.GetQuote(id)).Return(
        new Quotation(id, createdDateTime, new LineItem[]{}));

    var response = new HttpResponseMessage();

    var quote = new Quote(DefaultUriFactory.Instance, quoteEngine);
    quote.Get(
        id.ToString("N"),
        new HttpRequestMessage{Uri = new Uri("http://localhost/quote/")},
        response);

    Assert.AreEqual("public", response.Headers.CacheControl.ToString());
    Assert.AreEqual(expiryDateTime, response.Headers.Expires);
}
```

Add caching headers

```
public class Quote
{
    ...

    [HttpGet]
    public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage
        response)
    {
        ...

        response.StatusCode = HttpStatusCode.OK;
        response.Headers.CacheControl = new CacheControl {Public = true};
        response.Headers.Expires = quote.CreatedDateTime.AddDays(7.0).UtcDateTime;

        ...
    }
}
```

Media Types



Home page

```
<?xml version="1.0" encoding="utf-8"?>
<shop xmlns:rb="http://relations.restbucks.com/"
  xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <link rel="rb:rfq prefetch" type="application/vnd.restbucks+xml"
    href="request-for-quote/" />
</shop>
```

Quote

```
<?xml version="1.0" encoding="utf-8"?>
<shop xmlns:rb="http://relations.restbucks.com/"
  xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <items>
    <item>
      <description>coffee</description>
      <amount measure="g">125</amount>
      <price currency="GBP">1.25</price>
    </item>
  </items>
  <link rel="self" type="application/vnd.restbucks+xml"
    href="quote/68cff6e75a09474fa0098c9393aa6d4e" />
  <link rel="rb:order-form" type="application/vnd.restbucks+xml"
    href="order-form/68cff6e75a09474fa0098c9393aa6d4e" />
</shop>
```

Order form

```
<?xml version="1.0" encoding="utf-8"?>
<shop xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <model id="order" xmlns="http://www.w3.org/2002/xforms">
    <instance>
      <shop xml:base="http://win-cupsr6vr8g5/restbucks/"
        xmlns="http://schemas.restbucks.com/shop">
        <items>
          <item>
            <description>coffee</description>
            <amount measure="g">125</amount>
            <price currency="GBP">1.25</price>
          </item>
        </items>
        <link rel="self" type="application/vnd.restbucks+xml"
          href="quote/68cff6e75a09474fa0098c9393aa6d4e" />
      </shop>
    </instance>
    <submission
      resource="http://localhost:8081/orders/?c=12345&s=325"
      method="post"
      mediatype="application/vnd.restbucks+xml" />
  </model>
</shop>
```

Media type info set and formatter example

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <title type="text">Restbucks products and promotions</title>
  <id>urn:uuid:4aaf346c-1c9c-42cb-a006-5ff35d83c707</id>
  <updated>2010-11-29T04:38:09Z</updated>
  <author>
    <name>Product Catalog</name>
  </author>
  <generator>Product Catalog</generator>
  <link rel="self" href="http://localhost./updates" />
  <link rel="via" href="http://localhost./updates?p=3" />
  <link rel="prev-archive" href="http://localhost./updates?p=2" />
  <entry>
    ...
  </entry>
</feed>
```

Formatting

Atom10FeedFormatter

Info set

SyndicationFeed

Items: IEnumerable<SyndicationItem>

Restbucks media type library

```
<?xml version="1.0" encoding="utf-8"?>  
<shop xmlns:rb="http://relations.restbucks.com/"  
  xml:base="http://win-cupsr6vr8g5/restbucks/"  
  xmlns="http://schemas.restbucks.com/shop">  
  <link rel="rb:rfg prefetch" type="application/vnd.restbucks+xml"  
    href="request-for-quote/" />  
</shop>
```

Formatting

RestbucksMediaTypeProcessor

Assembler

Formatter

Infoset

Shop

Items: IEnumerable<Item>

Links: IEnumerable<Link>

Forms: IEnumerable<Form>

Home page

```
<?xml version="1.0" encoding="utf-8"?>
<shop xmlns:rb="http://relations.restbucks.com/"
  xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <link rel="rb:rfq prefetch" type="application/vnd.restbucks+xml"
    href="request-for-quote/" />
</shop>
```

```
var entityBody = new ShopBuilder(
    new Uri("http://win-cupsr6vr8gr/restbucks/"))
    .AddLink(new Link(
        new Uri("request-for-quote", UriKind.Relative),
        RestbucksMediaType.Value,
        LinkRelations.Rfq, LinkRelations.Prefetch)).Build();
```

Quote

```
<?xml version="1.0" encoding="utf-8"?>
<shop xmlns:rb="http://relations.restbucks.com/"
  xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <items>
    <item>
      <description>coffee</description>
      <amount measure="g">125</amount>
      <price currency="GBP">1.25</price>
```

```
var entityBody = new ShopBuilder(
    new Uri("http://win-cupsr6vr8gr/restbucks/"))
    .AddItem(new Item("coffee", new Amount("g", 125),
        new Cost("GBP", 1.25)))
    .AddLink(new Link(
        new Uri("quote/" + quotation.Id, UriKind.Relative),
        RestbucksMediaType.Value, LinkRelations.Self))
    .AddLink(new Link(
        new Uri("order-form/" + quotation.Id, UriKind.Relative),
        RestbucksMediaType.Value, LinkRelations.OrderForm))
    .Build();
```

Order form

```
<?xml version="1.0" encoding="utf-8"?>
<shop xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <model id="order" xmlns="http://www.w3.org/2002/xforms">
    <instance>
      <shop xml:base="http://win-cupsr6vr8g5/restbucks/"
```

```
var entityBody = new ShopBuilder(
    new Uri("http://win-cupsr6vr8gr/restbucks/"))
    .AddForm(new Form(FormSemantics.Order,
        new Uri("http://restbucks.com/orders"), "post",
        RestbucksMediaType.Value,
        new ShopBuilder(
            new Uri("http://win-cupsr6vr8gr/restbucks/"))
            .AddItem(new Item("coffee", new Amount("g", 125),
                new Cost("GBP", 1.25)))
            .AddLink(new Link(
                new Uri("quote/" + quotation.Id, UriKind.Relative),
                RestbucksMediaType.Value,
                LinkRelations.Self)).Build()))
    .Build();
```

Hypermedia



Link relationships



```
<link rel="payment"  
  href="https://payments.example.org/payments/1010"  
  type="application/xhtml+xml" />
```

```
<link rel="http://relations.restbucks.com/quote"  
  href="quote/68cff6e75a09474fa0098c9393aa6d4e"  
  type="application/vnd.restbucks+xml" />
```

```
<link rel="rb:rfq"  
  href="request-for-quote"  
  type="application/vnd.restbucks+xml"  
  xmlns:rb="http://relations.restbucks.com/" />
```



Registered

payment

Indicates a resource where payment is accepted

Extension

http://relations.restbucks.com/quote

Indicates that the linked resource is a quote

rb:rfq

Indicates a resource where a quote can be requested

Writing link relationships

```
[Test]
public void StringLinkRelationExample()
{
    var rel = new StringLinkRelation("self");
    Assert.AreEqual("self", rel.Value);
    Assert.AreEqual("self", rel.DisplayValue);
}
```

```
[Test]
public void UriLinkRelationExample()
{
    var rel = new UriLinkRelation(new Uri("http://relations.restbucks.com/quote"));
    Assert.AreEqual("http://relations.restbucks.com/quote", rel.Value);
    Assert.AreEqual("http://relations.restbucks.com/quote", rel.DisplayValue);
}
```

```
[Test]
public void CompactUriLinkRelationExample()
{
    var rel = new CompactUriLinkRelation("rb",
        new Uri("http://relations.restbucks.com/"), "rfq");
    Assert.AreEqual("http://relations.restbucks.com/rfq", rel.Value);
    Assert.AreEqual("rb:rfq", rel.DisplayValue);
}
```

Reading link relationships

```
[Test]
public void ShouldParseCompactUriIntoCompactUriLinkRelation()
{
    var rel = LinkRelation.Parse("rb:rfq", p => "http://relations.restbucks.com/");

    Assert.IsInstanceOf<CompactUriLinkRelation>(rel);
    Assert.AreEqual("rb:rfq", rel.DisplayValue);
    Assert.AreEqual("http://relations.restbucks.com/rfq", rel.Value);
}

public class LinksAssembler
{
    //Helper method
    public static Func<string, string> LookupNamespace(XElement element)
    {
        return prefix =>
        {
            var ns = element.GetNamespaceOfPrefix(prefix);
            return ns == null ? null : ns.NamespaceName;
        };
    }

    ...
}
```

Selecting links

```
<?xml version="1.0" encoding="utf-8"?>
<shop xmlns:rb="http://relations.restbucks.com/"
  xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <link rel="rb:rfq prefetch" type="application/vnd.restbucks+xml"
    href="request-for-quote/" />
</shop>
```

```
var prefetchLinks= (from l in entityBody.Links
                    where l.Rels.Contains(new StringLinkRelation("prefetch"),
                                           LinkRelationEqualityComparer.Instance)
                    select l);
```

```
var rfqLinks = (from l in entityBody.Links
                where l.Rels.Contains(new UriLinkRelation(
                    new Uri("http://relations.restbucks.com/rfq")),
                                       LinkRelationEqualityComparer.Instance)
                select l);
```


URIs everywhere

Routes

```
RouteTable.Routes.AddServiceRoute<Quote>("quote", configuration);  
RouteTable.Routes.AddServiceRoute<OrderForm>("order-form", configuration);
```

Methods

```
[HttpGet(UriTemplate = "{id}")]  
public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage  
    response)  
{  
    ...  
}
```

Links

```
return new ShopBuilder(new Uri("http://win-cupsr6vr8gr/restbucks/"))  
    .AddItem(new Item("coffee beans", new Amount("g", 250)))  
    .AddLink(new Link(  
        new Uri("quote/68cff6e75a09474fa0098c9393aa6d4e", UriKind.Relative),  
        RestbucksMediaType.Value, LinkRelations.Self))  
    .AddLink(new Link(  
        new Uri("order-form/68cff6e75a09474fa0098c9393aa6d4e", UriKind.Relative),  
        RestbucksMediaType.Value, LinkRelations.OrderForm))  
    .Build();
```

UriTemplateAttribute + UriFactory

```
[UriTemplate("quote", "{id}")]
public class Quote
{
    ...
}

[Test]
public void UriFactoryExample()
{
    var uriFactory = new UriFactory();
    uriFactory.Register<Quote>();

    Assert.AreEqual(
        new Uri("quote/1234", UriKind.Relative),
        uriFactory.CreateRelativeUri<Quote>(1234));

    Assert.AreEqual(
        new Uri("http://restbucks.com/quote/1234"),
        uriFactory.CreateAbsoluteUri<Quote>(new Uri("http://restbucks.com"), 1234));

    Assert.AreEqual(
        new Uri("http://restbucks.com/"),
        uriFactory.CreateBaseUri<Quote>(new Uri("http://restbucks.com/quote/1234")));
}
```

Registering resources at startup

```
public void RegisterResourcesFor(Assembly assembly)
{
    var register = typeof(UriFactory).GetMethod("Register",
        BindingFlags.Instance | BindingFlags.NonPublic);

    var resourceTypes = from t in assembly.GetTypes()
                        where t.GetCustomAttributes(typeof(UriTemplateAttribute),
                            false).Length > 0
                        select t;

    resourceTypes.ToList().ForEach(t =>
    {
        var genericMethod = register.MakeGenericMethod(new[] {t});
        genericMethod.Invoke(this, null);
    });
}
```

DRY URIs

```
[ServiceContract]
[UriTemplate("quote", "{id}")]
public class Quote
{
    private readonly UriFactory uriFactory;
    private readonly IQuotationEngine quoteEngine;

    ...

    [WebGet]
    public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage
        response)
    {
        ...
    }
}
```

```
var baseUri = uriFactory.CreateBaseUri<Quote>(request.Uri);

return new ShopBuilder(baseUri, quote.LineItems.Select(
    li => new LineItemToItem(li).Adapt()))
    .AddLink(new Link(uriFactory.CreateRelativeUri<Quote>(quote.Id),
        RestbucksMediaType.Value, LinkRelations.Self))
    .AddLink(new Link(uriFactory.CreateRelativeUri<OrderForm>(quote.Id),
        RestbucksMediaType.Value, LinkRelations.OrderForm)).Build();
}
```

Quote resource (complete code)

```
[ServiceContract]
[UriTemplate("quote", "{id}")]
public class Quote
{
    private readonly UriFactory uriFactory;
    private readonly IQotationEngine quoteEngine;

    public Quote(UriFactory uriFactory, IQotationEngine quoteEngine)
    {
        this.uriFactory = uriFactory;
        this.quoteEngine = quoteEngine;
    }

    [HttpGet]
    public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage response)
    {
        Quotation quote;
        try
        {
            quote = quoteEngine.GetQuote(new Guid(id));
        }
        catch (KeyNotFoundException)
        {
            response.StatusCode = HttpStatusCode.NotFound;
            return null;
        }

        response.StatusCode = HttpStatusCode.OK;
        response.Headers.CacheControl = new CacheControl { Public = true };
        response.Headers.Expires = quote.CreatedDateTime.AddDays(7.0).UtcDateTime;

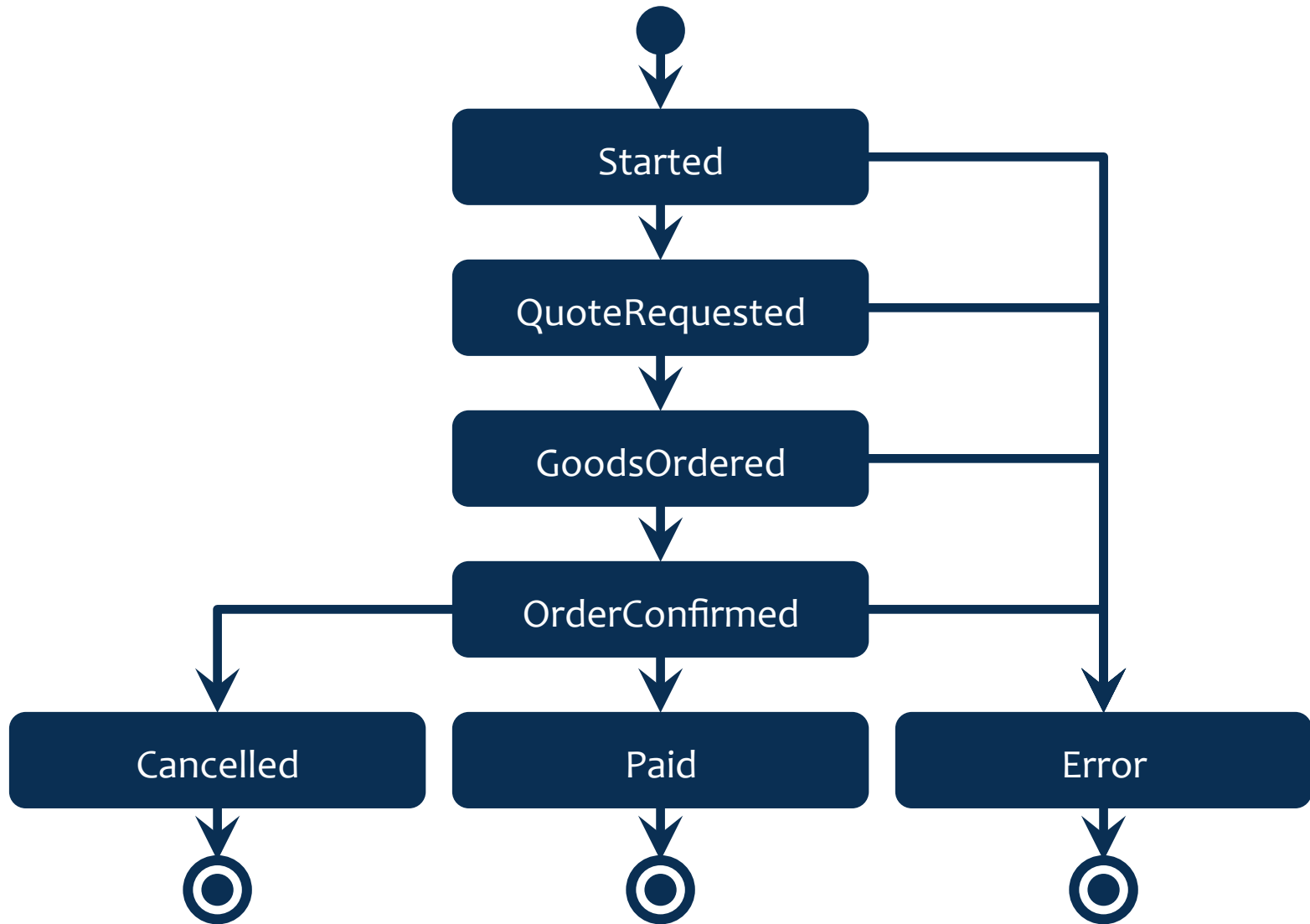
        var baseUri = uriFactory.CreateBaseUri<Quote>(request.Uri);

        return new ShopBuilder(baseUri, quote.LineItems.Select(
            li => new LineItemToItem(li).Adapt()))
            .AddLink(new Link(uriFactory.CreateRelativeUri<Quote>(quote.Id),
                RestbucksMediaType.Value, LinkRelations.Self))
            .AddLink(new Link(uriFactory.CreateRelativeUri<OrderForm>(quote.Id),
                RestbucksMediaType.Value, LinkRelations.OrderForm)).Build();
    }
}
```

Client-side Development



Client-side state machine



States and rules

```
public class StartedState : IState
{
    ...

    public IState Apply(IResponseHandlers handlers)
    {
        var rules = new Rules(
            When.IsTrue(IsUninitialized)
                .InvokeHandler(handlers.Get<UninitializedResponseHandler>)
                .UpdateContext(SetSemanticContext(SemanticContext.Started))
                .ReturnState(NewStartedState),
            When.IsTrue(IsStarted)
                .InvokeHandler(handlers.Get<StartedResponseHandler>)
                .UpdateContext(SetSemanticContext(SemanticContext.Rfq))
                .ReturnState(NewStartedState),
            When.IsTrue(IsRfc)
                .InvokeHandler(handlers.Get<RequestForQuoteFormResponseHandler>)
                .UpdateContext(ClearSemanticContext())
                .ReturnState(NewQuoteRequestedState));

        return rules.Evaluate(response, context);
    }

    ...
}
```


Helper methods

```
var rules = new Rules(  
    ...  
    When.IsTrue(IsRfc)  
        .InvokeHandler(handlers.Get<RequestForQuoteFormResponseHandler>())  
        .UpdateContext(ClearSemanticContext())  
        .ReturnState(NewQuoteRequestedState));  
  
private bool IsRfc()  
{  
    return response.IsSuccessStatusCode  
        && context.Get<string>(ApplicationContextKeys.SemanticContext)  
            .Equals(SemanticContext.Rfq);  
}  
  
private static Action<ApplicationContext> ClearSemanticContext()  
{  
    return c => c.Remove(ApplicationContextKeys.SemanticContext);  
}  
  
private static IState NewQuoteRequestedState(HttpResponseMessage response,  
    ApplicationContext context)  
{  
    return new QuoteRequestedState(response, context);  
}
```

Request-For-Quote

```
<?xml version="1.0" encoding="utf-8"?>
<shop xml:base="http://win-cupsr6vr8g5/restbucks/"
  xmlns="http://schemas.restbucks.com/shop">
  <model id="request-for-quote"
    schema="http://schemas.restbucks.com/shop"
    xmlns="http://www.w3.org/2002/xforms">
    <instance />
    <submission resource="quotes/"
      method="post" mediatype="application/vnd.restbucks+xml" />
  </model>
</shop>
```

Requesting a quote

```
public Result<HttpResponseMessage> Handle(HttpResponseMessage response,
    ApplicationContext context)
{
    var entityBody = response.Content
        .ReadAsObject<Shop>(RestbucksMediaTypeFormatter.Instance);
    var form = entityBody.Forms.First();

    var key = new EntityBodyKey(form.MediaType, form.Schema.ToString(),
        context.Get<string>(ApplicationContextKeys.ContextName));
    var formData = context.Get<Shop>(key);

    using (var client = clientProvider.CreateClient(entityBody.BaseUri))
    {
        var content = formData.ToContent(RestbucksMediaTypeFormatter.Instance);
        content.Headers.ContentType = new MediaTypeHeaderValue(
            RestbucksMediaType.Value);

        var request = new HttpRequestMessage(new HttpMethod(form.Method),
            form.Resource) {Content = content};
        var newResponse = client.Send(request);

        return new Result<HttpResponseMessage>(true, newResponse);
    }
}
```

Setting up the client

```
private static void Main(string[] args)
{
    var items = new ShopBuilder(null).AddItem(
        new Item("coffee", new Amount("g", 125)))
        .Build();

    var context = new ApplicationContext();
    context.Set(ApplicationContextKeys.EntryPointUri,
        new Uri("http://localhost/restbucks/shop/"));
    context.Set(new EntityBodyKey(RestbucksMediaType.Value,
        "http://schemas.restbucks.com/shop", SemanticContext.Rfq), items);

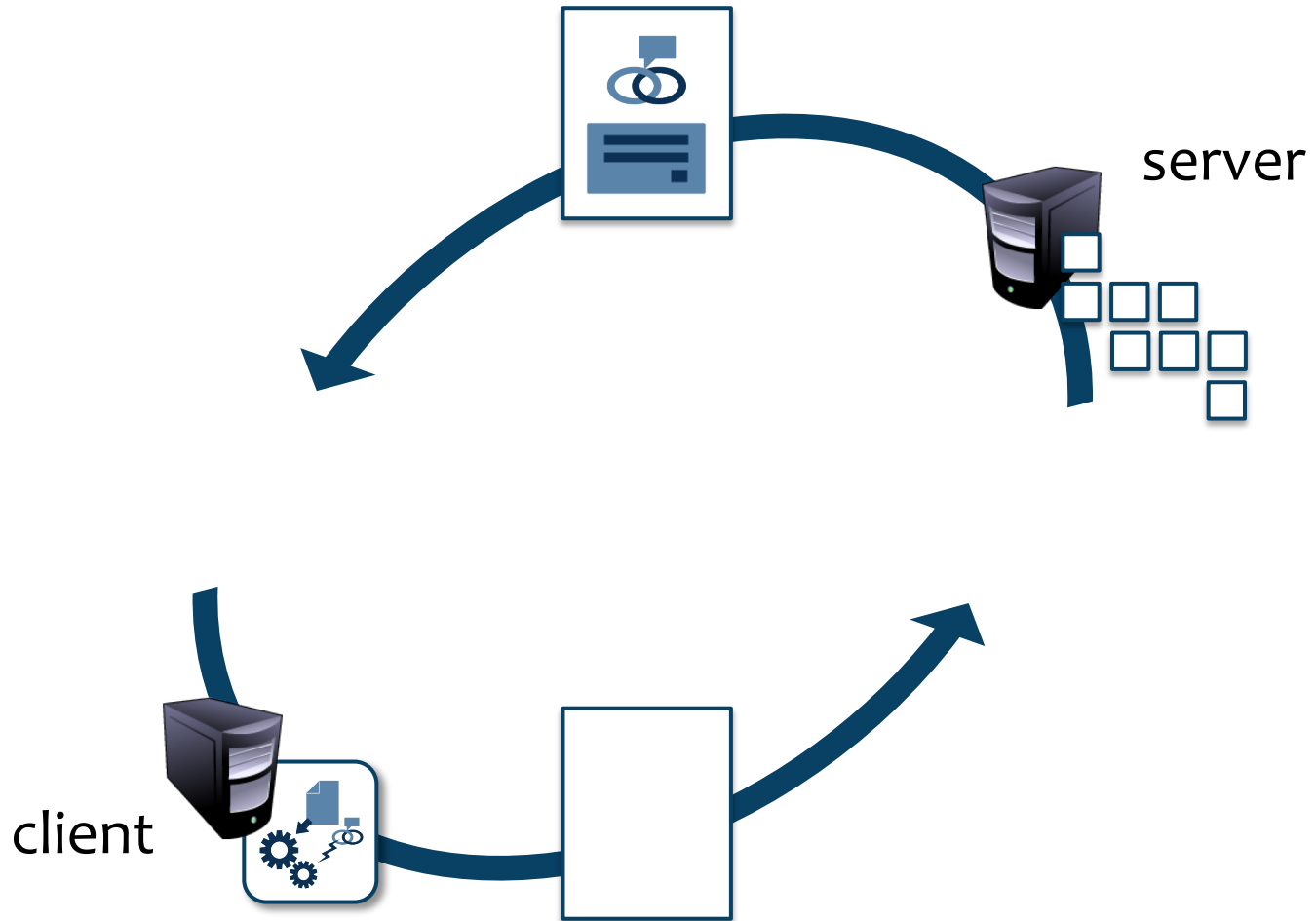
    var responseHandlers = new ResponseHandlers(HttpClientProvider.Instance);

    var state = new StartedState(null, context);
    var nextState = state.Apply(responseHandlers);

    while (!nextState.IsTerminalState)
    {
        nextState = nextState.Apply(responseHandlers);
    }

    Console.WriteLine("Finished");
    Console.ReadLine();
}
```

Summary

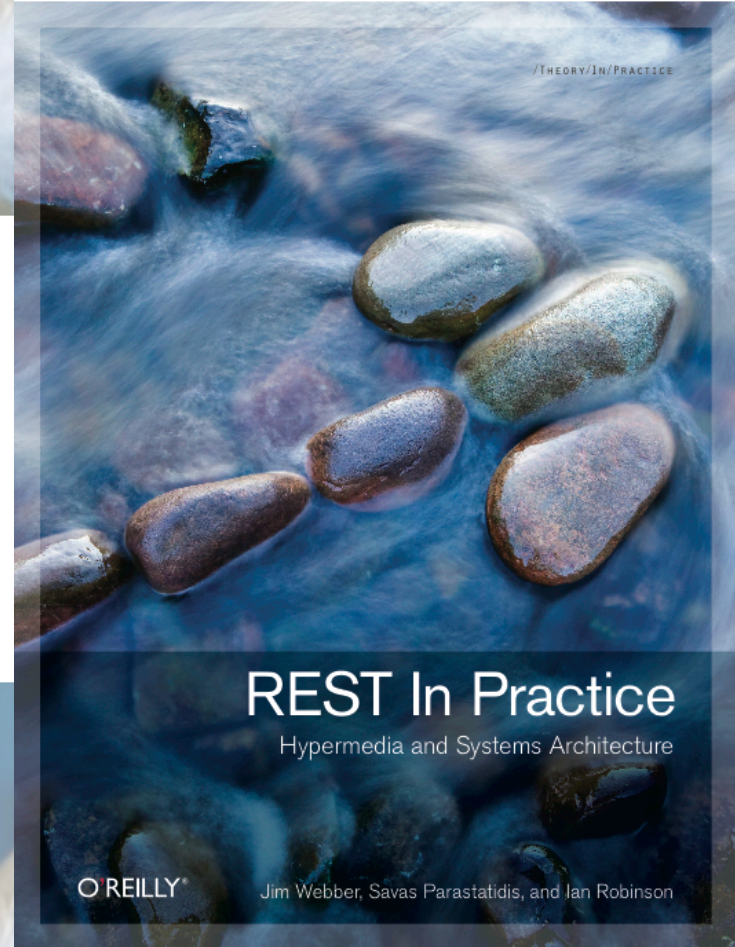


Thank you

<http://bit.ly/restqconlondon>

<http://ianSrobinson.com>
@ianSrobinson
ianSrobinson@gmail.com

ThoughtWorks®

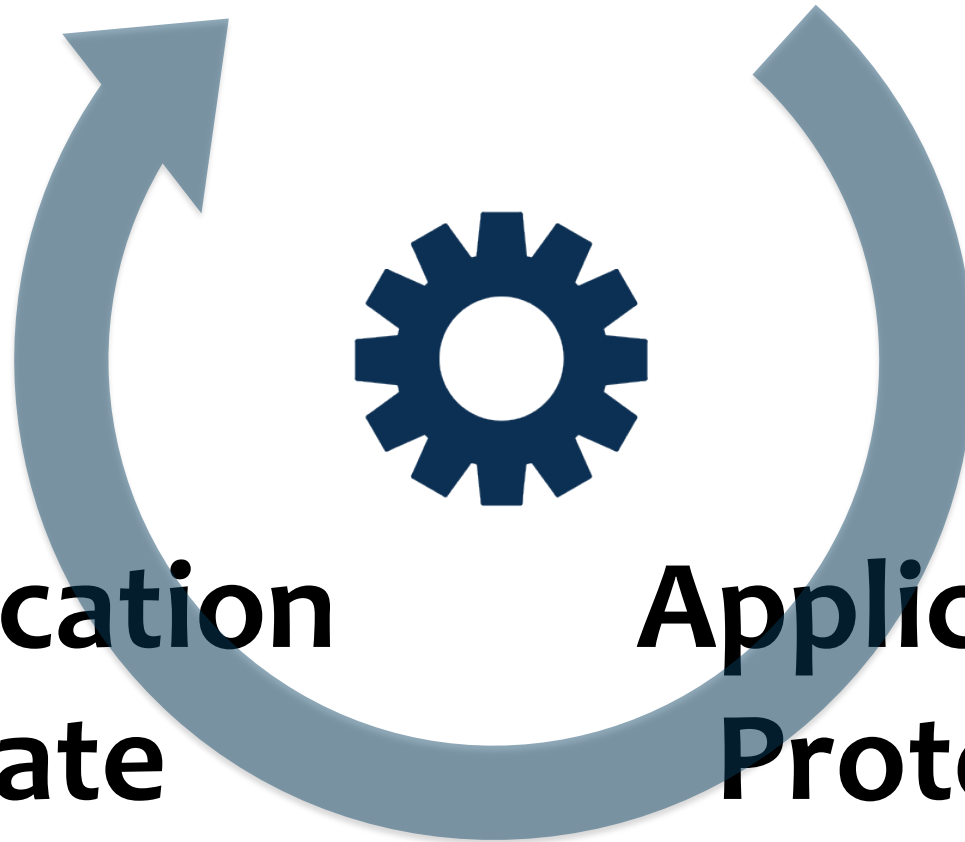


Releasing value

Application

**Application
State**

**Application
Protocol**



UriFactory

```
public class UriFactory
{
    private readonly IDictionary<Type, UriFactoryWorker> workers;

    public void Register<T>() where T : class
    {
        var attributes = typeof(T).GetCustomAttributes(
            typeof(UriTemplateAttribute), false);

        if (attributes.Length == 0)
        {
            throw new UriTemplateMissingException();
        }

        var worker = ((UriTemplateAttribute)attributes[0]).UriFactoryWorker;
        workers.Add(typeof(T), worker);
    }

    ...
}
```