

# Optimizing Your API for Ember Data

**Dan Gebhardt**

via Matthew Beale

## **Dan Gebhardt**

- Nice guy
- @dgeb on Github & Twitter
- Blogs at [cerebis.com/blog](http://cerebis.com/blog)

## **Matthew Beale**

- Also a nice guy
- @mixononic on Github & Twitter
- Blogs at [madhatted.com](http://madhatted.com)

“Trivial choices are the enemy”

Yehuda said that.

## **Ember Data makes some choices for you.**

- Canonical records
- In-memory store
- Customizable adapters & serializers
- Default adapter: DS.RESTAdapter
- Default serializer: DS.JSONSerializer

**And expects servers to follow conventions.**

- underscore\_naming
- Include root element
- id: 4, fk\_id: 4, fk\_ids: [4]
- Embedded data conventions
- REST Endpoints
- JSON
- ActiveRecord::Serializers gives you all of this (If you use Ruby).

## Edge of convention:



- Pagination
- Authentication (!!!)
- Sparse fieldsets
- Custom includes
- Polymorphism

Begin the shameful copy-paste code snippets...

## One-to-Many Relationships: Ember

```
App.Post = DS.Model.extend({  
  title: DS.attr('string'),  
  comments: DS.hasMany('App.Comment')  
});
```

```
App.Comment = DS.Model.extend({  
  body: DS.attr('string'),  
  post: DS.belongsTo('App.Post')  
});
```

Begin the shameful copy-paste code snippets...

## One-to-Many Relationships: AM::Serializers

```
class ApplicationSerializer < ActiveRecord::Serializer
  # sideload related data by default
  embed :ids, include: true
end
```

```
class PostSerializer < ApplicationSerializer
  attributes :title
  has_many :comments
end
```

```
class CommentSerializer < ApplicationSerializer
  attributes :body
  belongs_to :post
end
```



Begin the shameful copy-paste code snippets...

## One-to-Many Relationships: JSON

```
{
  post: {
    id: 1,
    title: 'Ember is Omakase',
    comment_ids: [4, 5, 6]
  },
  comments: [
    {id: 4, post_id: 1, body: 'delicious!'},
    {id: 5, post_id: 1, body: 'yuno turbolinks?'},
    {id: 6, post_id: 1, body: 'is that a tentacle?'}
  ]
}
```

## **Dan also goes over:**

- One-to-one
- One-to-none
- Many-to-many
- Many-to-none
- Embedded data (writable, ro)
- Custom pluralization
- Custom sideloading
- Bulk commits (this guy never stops!)

## **Matt will go over:**

- Custom keys
- Custom urls
- Custom transforms
- (But you should check out Dan's slides)

Continuing with the shameful copy-paste code snippets...

## Custom Keys: JSON

```
{
  post: {
    id: 1,
    titleOfPost: 'Ember is Omakase'
  }
}
```

Breaking with the shameful copy-paste code snippets!

## Custom Keys: Ember

```
App.Adapter.map( 'App.Post', {  
  title: { key: 'titleOfPost' }  
});
```

```
// This syntax is not the same as Dan's slide!!  
// His slide followed an old pattern. Use this.
```

```
// Lazy folks can call map on DS.RESTAdapter
```

Breaking with the shameful copy-paste code snippets!

## Custom URLs

```
// Again, this is a newer (I think better) syntax.
App.Adapter = DS.RESTAdapter.extend({
  namespace: 'ember'
});
App.Store = DS.Store.extend({
  adapter: 'App.Adapter'
});

// Then after the app is up

DS.defaultStore.find(Person, 1);
//=> /ember/people/1

// Along with namespace, you can specify a url
DS.defaultStore.set('adapter.url', 'http://api.ember.dev');

DS.defaultStore.find(Person, 1);
//=> http://api.ember.dev/ember/people/1
```

Back to the shameful copy-paste code snippets :-)

## Custom Transforms

```
Adapter.registerTransform('excitableString', {  
  serialize: function(value) {  
    return value + '!';  
  },  
  
  deserialize: function(value) {  
    return value.substring(0, value.length - 1);  
  }  
});
```

```
App.Greeting = DS.Model.extend({  
  message: DS.attr('excitableString')  
});
```

## MORE:

- Dan's slides: <https://speakerdeck.com/dgeb/optimizing-an-api-for-ember-data>
- Example codebase: [https://github.com/dgeb/ember\\_data\\_example](https://github.com/dgeb/ember_data_example)
- ActiveRecord::Serializers: [https://github.com/rails-api/active\\_model\\_serializers](https://github.com/rails-api/active_model_serializers)
- Stack Overflow! <http://stackoverflow.com/questions/tagged/ember-data>

Thanks!