

module

Sequel::Plugins::JsonSerializer

1. [lib/sequel/plugins/json_serializer.rb](#)

Parent: [Plugins](#)

The `json_serializer` plugin handles serializing entire `Sequel::Model` objects to JSON, as well as support for deserializing JSON directly into `Sequel::Model` objects. It requires the `json` library, and can work with either the pure ruby version or the C extension.

Basic Example:

```
album = Album[1]
album.to_json
# => '{"json_class"=>"Album", "id"=>1, "name"=>"RF", "artist_id"=>2}'
```

In addition, you can provide options to control the JSON output:

```
album.to_json(:only=>:name)
album.to_json(:except=>[:id, :artist_id])
# => '{"json_class"="Album", "name"=>"RF"}'
album.to_json(:include=>:artist)
# => '{"json_class": "Album", "id":1, "name": "RF", "artist_id":2,
      "artist":{"json_class": "Artist", "id":2, "name": "YJM"}'}
```

You can use a hash value with `:include` to pass options to associations:

```
album.to_json(:include=>{:artist=>{:only=>:name}})
# => '{"json_class": "Album", "id":1, "name": "RF", "artist_id":2,
      "artist":{"json_class": "Artist", "name": "YJM"}'}
```

You can specify the `:root` option to nest the JSON under the name of the model:

```
album.to_json(:root => true)
# => '{"album":{"id":1, "name": "RF", "artist_id":2}}'
```

In addition to creating JSON, this plugin also enables `Sequel::Model` objects to be automatically created when JSON is parsed:

```
json = album.to_json
album = JSON.parse(json)
```

In addition, you can update existing model objects directly from JSON using `from_json`:

```
album.from_json(json)
```

This works by parsing the JSON (which should return a hash), and then calling `set` with the returned hash.

Additionally, `to_json` also exists as a class and dataset method, both of which return all objects in the dataset:

```
Album.to_json  
Album.filter(:artist_id=>1).to_json(:include=>:tags)
```

Usage:

```
# Add JSON output capability to all model subclass instances (called before loading subclasses)  
Sequel::Model.plugin :json_serializer  
# Add JSON output capability to Album class instances  
Album.plugin :json_serializer
```

Methods

Public Class

1. [configure](#)

Classes and Modules

1. [Sequel::Plugins::JsonSerializer::ClassMethods](#)
2. [Sequel::Plugins::JsonSerializer::DatasetMethods](#)
3. [Sequel::Plugins::JsonSerializer::InstanceMethods](#)
4. [Sequel::Plugins::JsonSerializer::Literal](#)

Public Class methods

```
configure (model, opts={})
```

Set up the column readers to do deserialization and the column writers to save the value in `deserialized_values`.

[\[show source\]](#)

[Hanna RDoc template](#)