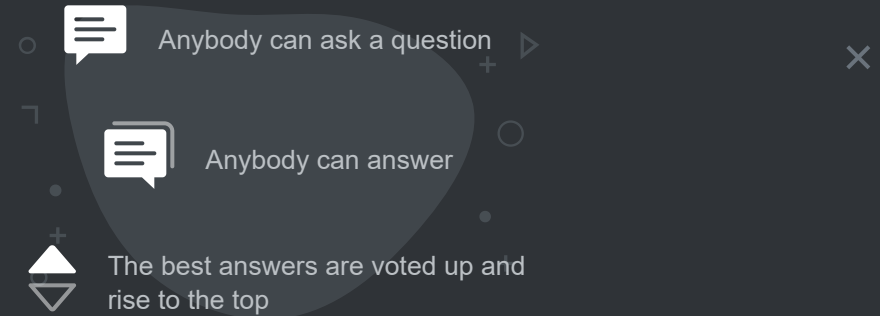


Software Engineering Stack Exchange is a question and answer site for professionals, academics, and students working within the systems development life cycle. It only takes a minute to sign up.

Sign up to join this community



Best practice loading entity with related entities

Asked 2 years, 7 months ago Active 2 years, 7 months ago Viewed 137 times



0



I am trying to figure what the best practice with loading data from web apis is.

Assume you have a model structure as per below:

```
class School {
  private string name;
  private List<Classroom> Rooms;
}

class Classroom {
  private string Type;
  private School school;
  private List<Student> Students;
}

class Student {
  private string Name;
  private int Age;
  private Classroom Room;
}
```

Is it better practice to load a single school entity and include all the related data (School + classrooms + students), then use the entity set at client side or is it better practice to just load the individual entities depending on what you need i.e. get School when school info needed, get Classroom when classroom needed so forth.

I can see pros and cons to both but keen to know if there is best practice.

[web-development](#)[entity-framework](#)

asked Nov 27 '17 at 23:19



[Aeseir](#)

201 ● 2 ● 8

2 Answers

[Active](#)[Oldest](#)[Votes](#)

Eager or Lazy Loading

3

If you know exactly how the data is used, you can easily decide, if eager loading (load everything at once) or lazy loading (do a second request if needed) gives better average performance.

Let the Consumer Decide



If your platform is more generic, so you don't have the information to optimize the data retrieval / transport, you should leave the decision to the API consumer.

Request Parameters

You could allow request parameters that specify, which related data is transmitted:

```
api/school/:name?with=rooms,rooms.students
```

GraphQL

Another way is [GraphQL](#), which allows you to specify the requested data in the request body.

```
api/school/:name
```

and the body to retrieve everything

```
{
  school {
    name
    rooms {
      type
      students {
        name
        age
      }
    }
  }
}
```

edited Nov 28 '17 at 17:25



Serhii Shushliapin

159 ● 5

answered Nov 28 '17 at 2:24



nibra

688 ● 3 ● 10

▲ The best practice here is going to depend entirely on what your specific requirements are.

2

▼ For example, your software might perform better if everything is loaded all at once. This is especially true if the data is being retrieved from a web service, where retrieving relatively large chunks of information is preferable to enduring the overhead of multiple service calls.



On the other hand, the related data might be infrequently accessed, which means that you would be wasting precious bandwidth and clock cycles retrieving data that is hardly ever used. In that case, it's better to defer retrieving the related data until it is actually needed.

Software development is a constant exercise in tradeoffs. Seldom is there a "right way" to do anything that applies to every possible situation.

answered Nov 27 '17 at 23:32



Robert Harvey

185k ● 50 ● 425 ● 635

Appreciate i might look into my setup then and see what might be best fit – Aeseir Nov 27 '17 at 23:33

