

Is JSON better than Partial Views for all use cases in ASP.NET MVC?

Asked 9 years, 2 months ago Active 1 year, 9 months ago Viewed 1k times

▲ There is a utility in jQuery called data template that can render the data through a template system.

4

Are data templates better than `RenderPartial` ?

▼

Can somebody explain their experience with it.

★

1

In my scenario it would be better because they make one less step for the developer and it is very customizable because it's easy to edit the template in the JS code instead of going to create new partial views.

🕒

So what is the better option in ASP.NET MVC?

jquery

json

asp.net-mvc

partial-views

datatemplate

edited Aug 22 '18 at 13:35



pnuts

52.6k ● 9 ● 71 ● 115

asked Apr 7 '11 at 6:50



delete my account

+1 for interesting question. I can see it being used for that. However, last i remember it is still in beta. – [Sergey Akopov](#) Apr 7 '11 at 7:06

The question as it stands is very subjective. The two are different technologies each with it's own pro's and con's. It would be better to understand the differences and benefits of each and consider them on each individual project rather than listen to opinion. You'll get more help if you rephrase the question. – [David Glenn](#) Apr 7 '11 at 8:24

4 Answers

Active

Oldest

Votes

▲

I am making the assumption you are working with MVC and are referring to partial views.

3

Partial view render on the server in one pass and are part of the page structure. Ajax calls will download the page first then inject the additional content later.

▼

Partial view give you SEO indexing. Ajax call give you intial page load speed.



Typically you want as much as possible in your initial page load then add dynamic content/feedback elements via ajax to enhance the user experience.



edited Apr 7 '11 at 10:04

answered Apr 7 '11 at 7:11



Slappy

3,807 ● 1 ● 23 ● 39



They are both useful and valid ways to render part of a page.

1

I'd say that partial views are more flexible because they do not depend on javascript and can help keep your site accessible if it needs to work with and without javascript. Remember you can load a PartialView into a `div` from an action method called via AJAX.



However, if the bias of your site is for a rich UI with few pages and most logic being driven by javascript then it would make sense to use the jQuery templates. It would be marginally quicker because of the smaller payload (though not much considering that you're normally simply omitting a small amount of markup).



Normally I go with partial views as there is a requirement for my javascript to be unobtrusive and it would be more work for little gain to use templates instead of pulling back partial views via AJAX.

answered Apr 7 '11 at 11:16



David Neale

14.3k ● 4 ● 53 ● 81



From my experience i would simply say Yes. Now let me explaining that in a bit:

1

The biggest issue with **RenderPartial** and **UpadtePanel** (in **WebForms**) is the simple fact that the actual html generation (i.e. output) is generated on server side, so all you got on client is to replace the updated **DOM** element with the new content.



What i dislike in this case is the **unnecessary** amount of data passing (sometimes in both directions) and the fact that there is a whole page rendering process going on on server side - which might translate to **performance**. Sure - for junior .net developers this way seem to be the perfect choice because it's straight forward and does not require additional **work** or knowledge, but i would avoid it whenever i can.



Now, the good thing with **jQuery Templates** is that the whole communication with the server is done through **JSON** (most likely) and the data footprint is significantly smaller. And it also gives you the **freedom** to easily modify the desired output without the need to change the server **logic** (e.g. redeploy your app on each change).

edited Mar 13 '18 at 19:33

answered Apr 7 '11 at 7:11



ljubomir

1,425 ● 21 ● 39

I believe he is referring to the MVC partial view and not the old MS AJAX Update Panel for Web Forms which was absolutely horrendous – Slappy Apr 7 '11 at 10:02

@Slappy i understood that, but wanted to make a point that both techniques are using somewhat same principles - server side rendering. And i agree that the latter was real mess - never liked it. – ljubomir Apr 7 '11 at 14:03

I disagree with what you said about junior developers. Learning about MVC and partial views will be learning curve for javascript developers who only know about how to use json. – Harry Mar 13 '18 at 18:18

@Harry thanks to your comment i have clarified my answer regarding junior .net developers. I wanted to make a point that using abstractions like UpdatePanel or RenderPartial seem simple and attractive (almost like magic), but this comes at a certain cost. And if one is aware of the trade-offs and still decides to go with it - then is ok. – ljubomir Mar 13 '18 at 19:36

- 1 I agree, but please don't forget that this thread was active almost 7 years ago. Things have changed a lot since then, especially the role of JavaScript in front-end development. Back at the days JSON and HTML templating was quite an new concept compared to todays technologies like Angular, React, Vue.js etc... – ljubomir Mar 14 '18 at 10:05



I did a quick search on Google and found a lot of good articles talking about this, and they will give you more information and better understand, just take a look at the first search result.

0

<http://www.google.com/search?sourceid=chrome&ie=UTF-8&q=jquery+template+vs+partial+views>



Also take a look at this question from stackoverflow:



[jQuery Templates vs Partial Views in ASP.NET MVC](#)

edited May 23 '17 at 11:55



Community ♦

1 ● 1

answered Apr 9 '11 at 1:38



Amr Elgarhy

54.8k ● 66 ● 170 ● 279