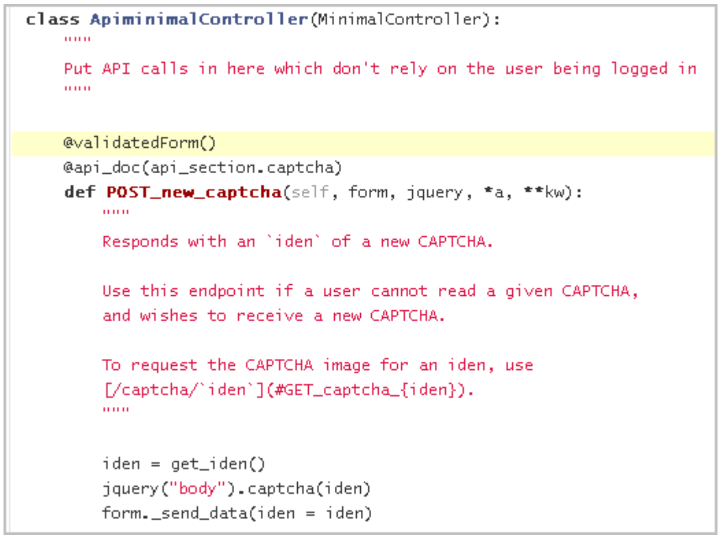
Documenting a REST API

Let's take a look an open source API that uses Representation State Transfer (REST) as the protocol.

Our example is Reddit, which like Wired Magazine, is owned by Conde Nast.

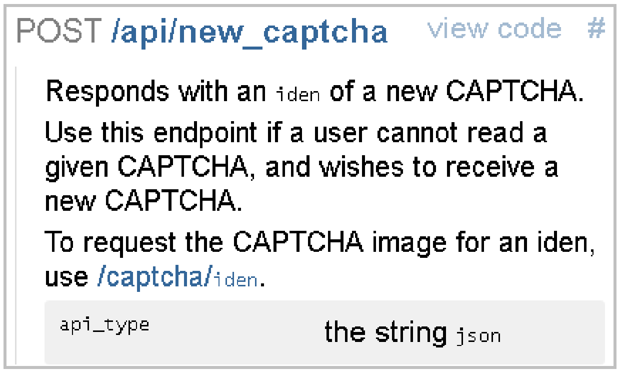
https://github.com/reddit/reddit/blob/master/r2/r2/controllers/api.py#L1401



```python
class ApiminimalController(MinimalController):
    """
    Put API calls in here which don't rely on the user being logged in
    """

    @validatedForm()
    @api_doc(api_section.captcha)
    def POST_new_captcha(self, form, jquery, *a, **kw):
        """
        Responds with an `iden` of a new CAPTCHA.

        Use this endpoint if a user cannot read a given CAPTCHA,
        and wishes to receive a new CAPTCHA.

        To request the CAPTCHA image for an iden, use
        [/captcha/`iden`](#GET_captcha_{iden}).
        """

        iden = get_iden()
        jquery("body").captcha(iden)
        form._send_data(iden = iden)
```

The comments for the function named `POST_new_captcha` are embedded in the Python source code inside three consecutive double-quote marks ("""), the Python delimiter for multi-line comments.

The API Reference documentation contains the comments for public functions at a URL separate from the source code URL - http://www.reddit.com/dev/api#POST_api_new_captcha



POST /api/new_captcha

Responds with an `iden` of a new CAPTCHA.

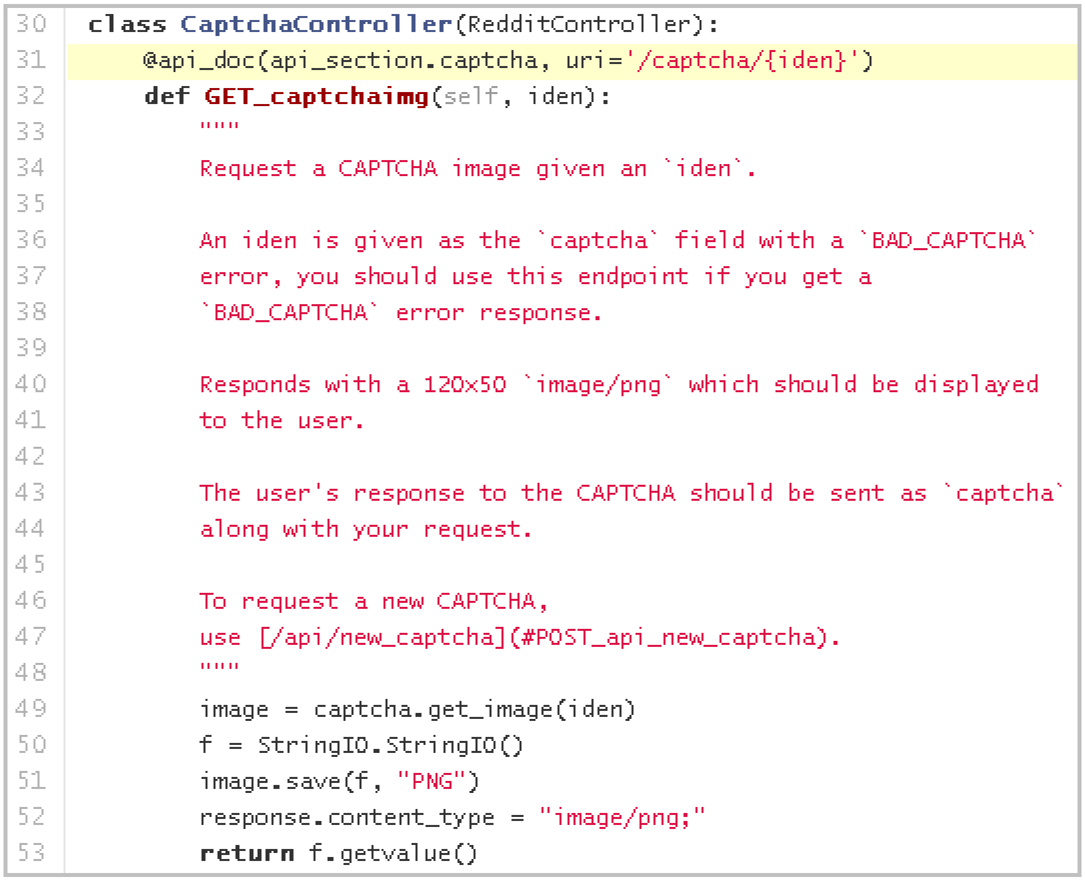
Use this endpoint if a user cannot read a given CAPTCHA, and wishes to receive a new CAPTCHA.

To request the CAPTCHA image for an iden, use /captcha/`iden`.

api_type	the string `json`

A technical writer can produce the REST API documentation provided she or he has access to an authoritative source of information, which might be the source code, a specification posted on a wiki, or answers the developer provides during an interview.

In this case, the technical writer reading the code sees that the response to the `GET_capchaimg` function is of the `content_type` `"image/png"` (Line 52).



```
30 class CaptchaController(RedditController):
31     @api_doc(api_section.captcha, uri='/captcha/{iden}')
32     def GET_captchaimg(self, iden):
33         """
34         Request a CAPTCHA image given an `iden`.
35 
36         An iden is given as the `captcha` field with a `BAD_CAPTCHA`
37         error, you should use this endpoint if you get a
38         `BAD_CAPTCHA` error response.
39 
40         Responds with a 120x50 `image/png` which should be displayed
41         to the user.
42 
43         The user's response to the CAPTCHA should be sent as `captcha`
44         along with your request.
45 
46         To request a new CAPTCHA,
47         use [/api/new_captcha](#POST_api_new_captcha).
48         """
49         image = captcha.get_image(iden)
50         f = StringIO.StringIO()
51         image.save(f, "PNG")
52         response.content_type = "image/png;"
53         return f.getvalue()
```

However, sometimes key information is not immediately available for the technical writer in the source code. For example, how would the technical writer know that the dimensions of the image are 120 x 50 pixels? It is possible that an Integrated Development Environment (IDE), such as Eclipse, would enable the technical writer to locate the definition of `image/png`, or that the technical writer could navigate the source code repository to the `image/` folder and get the properties of the files it contains. However, a developer would probably have to help the technical writer get set up during an orientation phase. Over time, the technical writer would become progressively more independent of the developer.