

implementation class serves as the actual logic and data access portion of the service, with the rest of the code delivering access to these methods. Each Facebook-specific tag, such as `<fb:TAG-NAME>`, will have a corresponding implementation method `fb_TAG_NAME` (e.g., the class method `fb_profile_pic` will implement the logic for the `<fb:profile-pic>` tag). Each standard HTML tag has a corresponding handler as well, named `tag_TAG_NAME`. These HTML handlers often let the data go through untouched, but often, FBML needs to make checks and do transforms even on “normal” HTML elements.

Let’s jump into the implementation of some of these tags, and then glue it all together. Each of these implementation methods accepts an `FBMLNode` returned from the FBML parser and returns some output HTML as a string. Here are example implementations for some direct HTML, data-display, and data-execution tags. Note that these listings use some functions not fully detailed here.

Implementing direct HTML tags in FBML

Example 6-26 contains the internal FBML implementation of the `` tag. The image tag’s implementation has some more logic, sometimes rewriting the image source URL to the URL of that image cached on Facebook’s servers. This demonstrates the power of FBML: an application stack can return markup very similar to the HTML used to support its own site, yet Facebook can enforce the behavior required by the Platform through purely technical means.

EXAMPLE 6-26. Implementation of the `fb:img` tag

```
class FBMLImplementation {
  public function __construct($flavor) {... }

  // <img>: example of direct HTML tag (section 4.3.1)
  public function tag_img($node) {

    // images are not allowed in some FBML contexts -
    // for example, the titles of feed stories
    $this->_flavor->check('images');

    // strip of transform attribute key-value pairs according to
    // rules in FBML
    $safe_attrs = $this->_html_rewriter->node_get_safe_attrs($node);
    if (isset($safe_attrs['src'])) {
      // may here rewrite image source to one on a Facebook CDN
      $safe_attrs['src'] = $this->_safe_image_url($safe_attrs['src']);
    }
    return $this->_html_rewriter->render_html_singleton_tag($node->
      get_tag_name(), $safe_attrs);
  }
}
```