# URL: https://flask.palletsprojects.com/en/2.0.x/patterns/appfactories/

Title: Application Factories — Flask Documentation (2.0.x)

Content:   
<!DOCTYPE html>  
  
<html lang="en">  
 <head>  
 <meta charset="utf-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Application Factories &#8212; Flask Documentation (2.0.x)</title>  
 <link rel="stylesheet" type="text/css" href="../../\_static/pygments.css" />  
 <link rel="stylesheet" type="text/css" href="../../\_static/flask.css" />  
 <link rel="stylesheet" type="text/css" href="../../\_static/tabs.css" />  
 <link rel="stylesheet" type="text/css" href="https://assets.readthedocs.org/static/css/badge\_only.css" />  
 <script data-url\_root="../../" id="documentation\_options" src="../../\_static/documentation\_options.js"></script>  
 <script src="../../\_static/jquery.js"></script>  
 <script src="../../\_static/underscore.js"></script>  
 <script src="../../\_static/doctools.js"></script>  
 <script src="../../\_static/tabs.js"></script>  
 <script async="async" src="https://assets.readthedocs.org/static/javascript/readthedocs-doc-embed.js"></script>  
 <link rel="canonical" href="https://flask.palletsprojects.com/en/2.1.x/patterns/appfactories/" />  
 <link rel="shortcut icon" href="../../\_static/flask-icon.png"/>  
 <link rel="index" title="Index" href="../../genindex/" />  
 <link rel="search" title="Search" href="../../search/" />  
 <link rel="next" title="Application Dispatching" href="../appdispatch/" />  
 <link rel="prev" title="Large Applications as Packages" href="../packages/" />   
   
<!-- RTD Extra Head -->  
  
<link rel="stylesheet" href="https://assets.readthedocs.org/static/css/readthedocs-doc-embed.css" type="text/css" />  
  
<script type="application/json" id="READTHEDOCS\_DATA">{"ad\_free": false, "api\_host": "https://readthedocs.org", "build\_date": "2022-03-28T19:28:32Z", "builder": "sphinx", "canonical\_url": null, "commit": "0305d31c", "docroot": "/docs/", "features": {"docsearch\_disabled": true}, "global\_analytics\_code": "UA-17997319-1", "language": "en", "page": "patterns/appfactories", "programming\_language": "py", "project": "flask", "proxied\_api\_host": "/\_", "source\_suffix": ".rst", "subprojects": {}, "theme": "flask", "user\_analytics\_code": "UA-162825511-1", "version": "2.0.x"}</script>  
  
<!--  
Using this variable directly instead of using `JSON.parse` is deprecated.  
The READTHEDOCS\_DATA global variable will be removed in the future.  
-->  
<script type="text/javascript">  
READTHEDOCS\_DATA = JSON.parse(document.getElementById('READTHEDOCS\_DATA').innerHTML);  
</script>  
  
<script type="text/javascript" src="https://assets.readthedocs.org/static/javascript/readthedocs-analytics.js" async="async"></script>  
  
<!-- end RTD <extrahead> -->  
</head><body>  
 <div class="related" role="navigation" aria-label="related navigation">  
 <h3>Navigation</h3>  
 <ul>  
 <li class="right" style="margin-right: 10px">  
 <a href="../../genindex/" title="General Index"  
 accesskey="I">index</a></li>  
 <li class="right" >  
 <a href="../../py-modindex/" title="Python Module Index"  
 >modules</a> |</li>  
 <li class="right" >  
 <a href="../appdispatch/" title="Application Dispatching"  
 accesskey="N">next</a> |</li>  
 <li class="right" >  
 <a href="../packages/" title="Large Applications as Packages"  
 accesskey="P">previous</a> |</li>  
 <li class="nav-item nav-item-0"><a href="../../">Flask Documentation (2.0.x)</a> &#187;</li>  
 <li class="nav-item nav-item-1"><a href="../" accesskey="U">Patterns for Flask</a> &#187;</li>  
 <li class="nav-item nav-item-this"><a href="">Application Factories</a></li>   
 </ul>  
 </div>   
  
 <div class="document">  
 <p class="version-warning"><strong>Warning:</strong> This is an old version. The latest stable version is <a href="../../../2.1.x/patterns/appfactories/">Version 2.1.x</a>.</p>  
 <div class="documentwrapper">  
 <div class="bodywrapper">  
 <div class="body" role="main">  
   
 <div class="section" id="application-factories">  
<h1>Application Factories<a class="headerlink" href="#application-factories" title="Permalink to this headline">Â¶</a></h1>  
<p>If you are already using packages and blueprints for your application  
(<a class="reference internal" href="../../blueprints/"><span class="doc">Modular Applications with Blueprints</span></a>) there are a couple of really nice ways to further improve  
the experience. A common pattern is creating the application object when  
the blueprint is imported. But if you move the creation of this object  
into a function, you can then create multiple instances of this app later.</p>  
<p>So why would you want to do this?</p>  
<ol class="arabic simple">  
<li><p>Testing. You can have instances of the application with different  
settings to test every case.</p></li>  
<li><p>Multiple instances. Imagine you want to run different versions of the  
same application. Of course you could have multiple instances with  
different configs set up in your webserver, but if you use factories,  
you can have multiple instances of the same application running in the  
same application process which can be handy.</p></li>  
</ol>  
<p>So how would you then actually implement that?</p>  
<div class="section" id="basic-factories">  
<h2>Basic Factories<a class="headerlink" href="#basic-factories" title="Permalink to this headline">Â¶</a></h2>  
<p>The idea is to set up the application in a function. Like this:</p>  
<div class="highlight-default notranslate"><div class="highlight"><pre><span></span><span class="k">def</span> <span class="nf">create\_app</span><span class="p">(</span><span class="n">config\_filename</span><span class="p">):</span>  
 <span class="n">app</span> <span class="o">=</span> <span class="n">Flask</span><span class="p">(</span><span class="vm">\_\_name\_\_</span><span class="p">)</span>  
 <span class="n">app</span><span class="o">.</span><span class="n">config</span><span class="o">.</span><span class="n">from\_pyfile</span><span class="p">(</span><span class="n">config\_filename</span><span class="p">)</span>  
  
 <span class="kn">from</span> <span class="nn">yourapplication.model</span> <span class="kn">import</span> <span class="n">db</span>  
 <span class="n">db</span><span class="o">.</span><span class="n">init\_app</span><span class="p">(</span><span class="n">app</span><span class="p">)</span>  
  
 <span class="kn">from</span> <span class="nn">yourapplication.views.admin</span> <span class="kn">import</span> <span class="n">admin</span>  
 <span class="kn">from</span> <span class="nn">yourapplication.views.frontend</span> <span class="kn">import</span> <span class="n">frontend</span>  
 <span class="n">app</span><span class="o">.</span><span class="n">register\_blueprint</span><span class="p">(</span><span class="n">admin</span><span class="p">)</span>  
 <span class="n">app</span><span class="o">.</span><span class="n">register\_blueprint</span><span class="p">(</span><span class="n">frontend</span><span class="p">)</span>  
  
 <span class="k">return</span> <span class="n">app</span>  
</pre></div>  
</div>  
<p>The downside is that you cannot use the application object in the blueprints  
at import time. You can however use it from within a request. How do you  
get access to the application with the config? Use  
<a class="reference internal" href="../../api/#flask.current\_app" title="flask.current\_app"><code class="xref py py-data docutils literal notranslate"><span class="pre">current\_app</span></code></a>:</p>  
<div class="highlight-default notranslate"><div class="highlight"><pre><span></span><span class="kn">from</span> <span class="nn">flask</span> <span class="kn">import</span> <span class="n">current\_app</span><span class="p">,</span> <span class="n">Blueprint</span><span class="p">,</span> <span class="n">render\_template</span>  
<span class="n">admin</span> <span class="o">=</span> <span class="n">Blueprint</span><span class="p">(</span><span class="s1">&#39;admin&#39;</span><span class="p">,</span> <span class="vm">\_\_name\_\_</span><span class="p">,</span> <span class="n">url\_prefix</span><span class="o">=</span><span class="s1">&#39;/admin&#39;</span><span class="p">)</span>  
  
<span class="nd">@admin</span><span class="o">.</span><span class="n">route</span><span class="p">(</span><span class="s1">&#39;/&#39;</span><span class="p">)</span>  
<span class="k">def</span> <span class="nf">index</span><span class="p">():</span>  
 <span class="k">return</span> <span class="n">render\_template</span><span class="p">(</span><span class="n">current\_app</span><span class="o">.</span><span class="n">config</span><span class="p">[</span><span class="s1">&#39;INDEX\_TEMPLATE&#39;</span><span class="p">])</span>  
</pre></div>  
</div>  
<p>Here we look up the name of a template in the config.</p>  
</div>  
<div class="section" id="factories-extensions">  
<h2>Factories &amp; Extensions<a class="headerlink" href="#factories-extensions" title="Permalink to this headline">Â¶</a></h2>  
<p>Itâs preferable to create your extensions and app factories so that the  
extension object does not initially get bound to the application.</p>  
<p>Using <a class="reference external" href="https://flask-sqlalchemy.palletsprojects.com/">Flask-SQLAlchemy</a>,  
as an example, you should not do something along those lines:</p>  
<div class="highlight-default notranslate"><div class="highlight"><pre><span></span><span class="k">def</span> <span class="nf">create\_app</span><span class="p">(</span><span class="n">config\_filename</span><span class="p">):</span>  
 <span class="n">app</span> <span class="o">=</span> <span class="n">Flask</span><span class="p">(</span><span class="vm">\_\_name\_\_</span><span class="p">)</span>  
 <span class="n">app</span><span class="o">.</span><span class="n">config</span><span class="o">.</span><span class="n">from\_pyfile</span><span class="p">(</span><span class="n">config\_filename</span><span class="p">)</span>  
  
 <span class="n">db</span> <span class="o">=</span> <span class="n">SQLAlchemy</span><span class="p">(</span><span class="n">app</span><span class="p">)</span>  
</pre></div>  
</div>  
<p>But, rather, in model.py (or equivalent):</p>  
<div class="highlight-default notranslate"><div class="highlight"><pre><span></span><span class="n">db</span> <span class="o">=</span> <span class="n">SQLAlchemy</span><span class="p">()</span>  
</pre></div>  
</div>  
<p>and in your application.py (or equivalent):</p>  
<div class="highlight-default notranslate"><div class="highlight"><pre><span></span><span class="k">def</span> <span class="nf">create\_app</span><span class="p">(</span><span class="n">config\_filename</span><span class="p">):</span>  
 <span class="n">app</span> <span class="o">=</span> <span class="n">Flask</span><span class="p">(</span><span class="vm">\_\_name\_\_</span><span class="p">)</span>  
 <span class="n">app</span><span class="o">.</span><span class="n">config</span><span class="o">.</span><span class="n">from\_pyfile</span><span class="p">(</span><span class="n">config\_filename</span><span class="p">)</span>  
  
 <span class="kn">from</span> <span class="nn">yourapplication.model</span> <span class="kn">import</span> <span class="n">db</span>  
 <span class="n">db</span><span class="o">.</span><span class="n">init\_app</span><span class="p">(</span><span class="n">app</span><span class="p">)</span>  
</pre></div>  
</div>  
<p>Using this design pattern, no application-specific state is stored on the  
extension object, so one extension object can be used for multiple apps.  
For more information about the design of extensions refer to <a class="reference internal" href="../../extensiondev/"><span class="doc">Flask Extension Development</span></a>.</p>  
</div>  
<div class="section" id="using-applications">  
<h2>Using Applications<a class="headerlink" href="#using-applications" title="Permalink to this headline">Â¶</a></h2>  
<p>To run such an application, you can use the <strong class="command">flask</strong> command:</p>  
<div class="sphinx-tabs docutils container">  
<div aria-label="Tabbed content" class="closeable" role="tablist"><button aria-controls="panel-0-QmFzaA==" aria-selected="true" class="sphinx-tabs-tab group-tab" id="tab-0-QmFzaA==" name="QmFzaA==" role="tab" tabindex="0">Bash</button><button aria-controls="panel-0-RmlzaA==" aria-selected="false" class="sphinx-tabs-tab group-tab" id="tab-0-RmlzaA==" name="RmlzaA==" role="tab" tabindex="-1">Fish</button><button aria-controls="panel-0-Q01E" aria-selected="false" class="sphinx-tabs-tab group-tab" id="tab-0-Q01E" name="Q01E" role="tab" tabindex="-1">CMD</button><button aria-controls="panel-0-UG93ZXJzaGVsbA==" aria-selected="false" class="sphinx-tabs-tab group-tab" id="tab-0-UG93ZXJzaGVsbA==" name="UG93ZXJzaGVsbA==" role="tab" tabindex="-1">Powershell</button></div><div aria-labelledby="tab-0-QmFzaA==" class="sphinx-tabs-panel group-tab" id="panel-0-QmFzaA==" name="QmFzaA==" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>$ export FLASK\_APP=myapp  
$ flask run  
</pre></div>  
</div>  
</div><div aria-labelledby="tab-0-RmlzaA==" class="sphinx-tabs-panel group-tab" hidden="true" id="panel-0-RmlzaA==" name="RmlzaA==" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>$ set -x FLASK\_APP myapp  
$ flask run  
</pre></div>  
</div>  
</div><div aria-labelledby="tab-0-Q01E" class="sphinx-tabs-panel group-tab" hidden="true" id="panel-0-Q01E" name="Q01E" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>&gt; set FLASK\_APP=myapp  
&gt; flask run  
</pre></div>  
</div>  
</div><div aria-labelledby="tab-0-UG93ZXJzaGVsbA==" class="sphinx-tabs-panel group-tab" hidden="true" id="panel-0-UG93ZXJzaGVsbA==" name="UG93ZXJzaGVsbA==" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>&gt; $env:FLASK\_APP = &quot;myapp&quot;  
&gt; flask run  
</pre></div>  
</div>  
</div></div>  
<p>Flask will automatically detect the factory (<code class="docutils literal notranslate"><span class="pre">create\_app</span></code> or <code class="docutils literal notranslate"><span class="pre">make\_app</span></code>)  
in <code class="docutils literal notranslate"><span class="pre">myapp</span></code>. You can also pass arguments to the factory like this:</p>  
<div class="sphinx-tabs docutils container">  
<div aria-label="Tabbed content" class="closeable" role="tablist"><button aria-controls="panel-1-QmFzaA==" aria-selected="true" class="sphinx-tabs-tab group-tab" id="tab-1-QmFzaA==" name="QmFzaA==" role="tab" tabindex="0">Bash</button><button aria-controls="panel-1-RmlzaA==" aria-selected="false" class="sphinx-tabs-tab group-tab" id="tab-1-RmlzaA==" name="RmlzaA==" role="tab" tabindex="-1">Fish</button><button aria-controls="panel-1-Q01E" aria-selected="false" class="sphinx-tabs-tab group-tab" id="tab-1-Q01E" name="Q01E" role="tab" tabindex="-1">CMD</button><button aria-controls="panel-1-UG93ZXJzaGVsbA==" aria-selected="false" class="sphinx-tabs-tab group-tab" id="tab-1-UG93ZXJzaGVsbA==" name="UG93ZXJzaGVsbA==" role="tab" tabindex="-1">Powershell</button></div><div aria-labelledby="tab-1-QmFzaA==" class="sphinx-tabs-panel group-tab" id="panel-1-QmFzaA==" name="QmFzaA==" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>$ export FLASK\_APP=&quot;myapp:create\_app(&#39;dev&#39;)&quot;  
$ flask run  
</pre></div>  
</div>  
</div><div aria-labelledby="tab-1-RmlzaA==" class="sphinx-tabs-panel group-tab" hidden="true" id="panel-1-RmlzaA==" name="RmlzaA==" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>$ set -x FLASK\_APP &quot;myapp:create\_app(&#39;dev&#39;)&quot;  
$ flask run  
</pre></div>  
</div>  
</div><div aria-labelledby="tab-1-Q01E" class="sphinx-tabs-panel group-tab" hidden="true" id="panel-1-Q01E" name="Q01E" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>&gt; set FLASK\_APP=&quot;myapp:create\_app(&#39;dev&#39;)&quot;  
&gt; flask run  
</pre></div>  
</div>  
</div><div aria-labelledby="tab-1-UG93ZXJzaGVsbA==" class="sphinx-tabs-panel group-tab" hidden="true" id="panel-1-UG93ZXJzaGVsbA==" name="UG93ZXJzaGVsbA==" role="tabpanel" tabindex="0"><div class="highlight-text notranslate"><div class="highlight"><pre><span></span>&gt; $env:FLASK\_APP = &quot;myapp:create\_app(&#39;dev&#39;)&quot;  
&gt; flask run  
</pre></div>  
</div>  
</div></div>  
<p>Then the <code class="docutils literal notranslate"><span class="pre">create\_app</span></code> factory in <code class="docutils literal notranslate"><span class="pre">myapp</span></code> is called with the string  
<code class="docutils literal notranslate"><span class="pre">'dev'</span></code> as the argument. See <a class="reference internal" href="../../cli/"><span class="doc">Command Line Interface</span></a> for more detail.</p>  
</div>  
<div class="section" id="factory-improvements">  
<h2>Factory Improvements<a class="headerlink" href="#factory-improvements" title="Permalink to this headline">Â¶</a></h2>  
<p>The factory function above is not very clever, but you can improve it.  
The following changes are straightforward to implement:</p>  
<ol class="arabic simple">  
<li><p>Make it possible to pass in configuration values for unit tests so that  
you donât have to create config files on the filesystem.</p></li>  
<li><p>Call a function from a blueprint when the application is setting up so  
that you have a place to modify attributes of the application (like  
hooking in before/after request handlers etc.)</p></li>  
<li><p>Add in WSGI middlewares when the application is being created if necessary.</p></li>  
</ol>  
</div>  
</div>  
  
  
 <div class="clearer"></div>  
 </div>  
 </div>  
 </div>  
 <span id="sidebar-top"></span>  
 <div class="sphinxsidebar" role="navigation" aria-label="main navigation">  
 <div class="sphinxsidebarwrapper">  
   
   
 <p class="logo"><a href="../../">  
 <img class="logo" src="../../\_static/flask-icon.png" alt="Logo"/>  
 </a></p>  
   
  
 <h3>Contents</h3>  
 <ul>  
<li><a class="reference internal" href="#">Application Factories</a><ul>  
<li><a class="reference internal" href="#basic-factories">Basic Factories</a></li>  
<li><a class="reference internal" href="#factories-extensions">Factories &amp; Extensions</a></li>  
<li><a class="reference internal" href="#using-applications">Using Applications</a></li>  
<li><a class="reference internal" href="#factory-improvements">Factory Improvements</a></li>  
</ul>  
</li>  
</ul>  
<h3>Navigation</h3>  
<ul>  
 <li><a href="../../">Overview</a>  
 <ul>  
 <li><a href="../">Patterns for Flask</a>  
 <ul>  
 <li>Previous: <a href="../packages/" title="previous chapter">Large Applications as Packages</a>  
 <li>Next: <a href="../appdispatch/" title="next chapter">Application Dispatching</a></ul>  
 </li>  
 </ul>  
 </li>  
</ul>  
<div id="searchbox" style="display: none" role="search">  
 <h3 id="searchlabel">Quick search</h3>  
 <div class="searchformwrapper">  
 <form class="search" action="../../search/" method="get">  
 <input type="text" name="q" aria-labelledby="searchlabel" autocomplete="off" autocorrect="off" autocapitalize="off" spellcheck="false"/>  
 <input type="submit" value="Go" />  
 </form>  
 </div>  
</div>  
<script>$('#searchbox').show(0);</script><div id="ethical-ad-placement"></div>  
 </div>  
 </div>  
 <div class="clearer"></div>  
 </div>  
   
 <div class="footer" role="contentinfo">  
 &#169; Copyright 2010 Pallets.  
 Created using <a href="https://www.sphinx-doc.org/">Sphinx</a> 4.4.0.  
 </div>  
 <script src="../../\_static/version\_warning\_offset.js"></script>  
  
 </body>  
</html>

File Path: N/A

Links: ., ., /, ., ., /, g, e, n, i, n, d, e, x, /, ,, ., ., /, ., ., /, p, y, -, m, o, d, i, n, d, e, x, /, ,, ., ., /, a, p, p, d, i, s, p, a, t, c, h, /, ,, ., ., /, p, a, c, k, a, g, e, s, /, ,, ., ., /, ., ., /, ,, ., ., /, ,, ., ., /, ., ., /, ., ., /, 2, ., 1, ., x, /, p, a, t, t, e, r, n, s, /, a, p, p, f, a, c, t, o, r, i, e, s, /, ,, #, a, p, p, l, i, c, a, t, i, o, n, -, f, a, c, t, o, r, i, e, s, ,, ., ., /, ., ., /, b, l, u, e, p, r, i, n, t, s, /, ,, #, b, a, s, i, c, -, f, a, c, t, o, r, i, e, s, ,, ., ., /, ., ., /, a, p, i, /, #, f, l, a, s, k, ., c, u, r, r, e, n, t, \_, a, p, p, ,, #, f, a, c, t, o, r, i, e, s, -, e, x, t, e, n, s, i, o, n, s, ,, h, t, t, p, s, :, /, /, f, l, a, s, k, -, s, q, l, a, l, c, h, e, m, y, ., p, a, l, l, e, t, s, p, r, o, j, e, c, t, s, ., c, o, m, /, ,, ., ., /, ., ., /, e, x, t, e, n, s, i, o, n, d, e, v, /, ,, #, u, s, i, n, g, -, a, p, p, l, i, c, a, t, i, o, n, s, ,, ., ., /, ., ., /, c, l, i, /, ,, #, f, a, c, t, o, r, y, -, i, m, p, r, o, v, e, m, e, n, t, s, ,, ., ., /, ., ., /, ,, #, ,, #, b, a, s, i, c, -, f, a, c, t, o, r, i, e, s, ,, #, f, a, c, t, o, r, i, e, s, -, e, x, t, e, n, s, i, o, n, s, ,, #, u, s, i, n, g, -, a, p, p, l, i, c, a, t, i, o, n, s, ,, #, f, a, c, t, o, r, y, -, i, m, p, r, o, v, e, m, e, n, t, s, ,, ., ., /, ., ., /, ,, ., ., /, ,, ., ., /, p, a, c, k, a, g, e, s, /, ,, ., ., /, a, p, p, d, i, s, p, a, t, c, h, /, ,, h, t, t, p, s, :, /, /, w, w, w, ., s, p, h, i, n, x, -, d, o, c, ., o, r, g, /, ,, ., ., /, ., ., /, \_, s, t, a, t, i, c, /, f, l, a, s, k, -, i, c, o, n, ., p, n, g