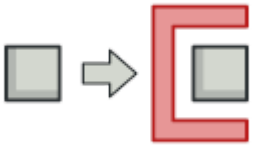

Proxy in Java

Proxy is a structural design pattern that provides an object that acts as a substitute for a real service object used by a client. A proxy receives client requests, does some work (access control, caching, etc.) and then passes the request to a service object.

The proxy object has the same interface as a service, which makes it interchangeable with a real object when passed to a client.

Learn more about Proxy →

Navigation

Complexity: ★★☆

Usage examples: While the Proxy pattern isn't a frequent guest in most Java applications, it's still very handy in some special cases. It's irreplaceable when you want to add some additional behaviors to an object of some existing class without changing the client code.

Some examples of proxies in standard Java libraries:

- `java.lang.reflect.Proxy`
- `java.rmi.*`
- `javax.ejb.EJB` (see comments)
- `javax.inject.Inject` (see comments)
- `javax.persistence.PersistenceContext`

Identification: Proxies delegate all of the real work to some other object. Each proxy method should, in the end, refer to a service object unless the proxy is a subclass of a service.

Caching proxy

In this example, the Proxy pattern helps to implement the lazy initialization and caching to an inefficient 3rd-party YouTube integration library.

Proxy is invaluable when you have to add some additional behaviors to a class which code you can't change.

some_cool_media_library

some_cool_media_library/ThirdPartyYouTubeLib.java: Remote service interface

```
package refactoring_guru.proxy.example.some_cool_media_library;

import java.util.HashMap;

public interface ThirdPartyYouTubeLib {
    HashMap<String, Video> popularVideos();

    Video getVideo(String videoId);
}
```

Implementation

```
package refactoring_guru.proxy.example.some_cool_media_library;

import java.util.HashMap;

public class ThirdPartyYouTubeClass implements ThirdPartyYouTubeLib {

    @Override
    public HashMap<String, Video> popularVideos() {
        connectToServer("http://www.youtube.com");
        return getRandomVideos();
    }

    @Override
    public Video getVideo(String videoId) {
        connectToServer("http://www.youtube.com/" + videoId);
        return getSomeVideo(videoId);
    }

    // ----------------------------------------------------------------------
    // Fake methods to simulate network activity. They as slow as a real life.

    private int random(int min, int max) {
        return min + (int) (Math.random() * ((max - min) + 1));
    }

    private void experienceNetworkLatency() {
        int randomLatency = random(5, 10);
        for (int i = 0; i < randomLatency; i++) {
            try {
                Thread.sleep(100);
            } catch (InterruptedException ex) {
                ex.printStackTrace();
            }
        }
    }

    private void connectToServer(String server) {
        System.out.print("Connecting to " + server + "... ");
        experienceNetworkLatency();
        System.out.print("Connected!" + "\n");
    }

    private HashMap<String, Video> getRandomVideos() {
        System.out.print("Downloading populars... ");

        experienceNetworkLatency();
        HashMap<String, Video> hmap = new HashMap<String, Video>();
```

```
        hmap.put("dancesvideoo", new Video("asdfas3ffasd", "Dancing video.mpq"));
        hmap.put("dlsdk5jfslaf", new Video("dlsdk5jfslaf", "Barcelona vs RealM.mov"));
        hmap.put("3sdfgsd1j333", new Video("3sdfgsd1j333", "Programing lesson#1.avi"));

        System.out.print("Done!" + "\n");
        return hmap;
    }

    private Video getSomeVideo(String videoId) {
        System.out.print("Downloading video... ");

        experienceNetworkLatency();
        Video video = new Video(videoId, "Some video title");

        System.out.print("Done!" + "\n");
        return video;
    }

}
```

some_cool_media_library/Video.java: Video file

```
package refactoring_guru.proxy.example.some_cool_media_library;

public class Video {
    public String id;
    public String title;
    public String data;

    Video(String id, String title) {
        this.id = id;
        this.title = title;
        this.data = "Random video.";
    }
}
```

proxy

proxy/YouTubeCacheProxy.java: Caching proxy

```
package refactoring_guru.proxy.example.proxy;
```

```
import refactoring_guru.proxy.example.some_cool_media_library.Video;

import java.util.HashMap;

public class YouTubeCacheProxy implements ThirdPartyYouTubeLib {
    private ThirdPartyYouTubeLib youtubeService;
    private HashMap<String, Video> cachePopular = new HashMap<String, Video>();
    private HashMap<String, Video> cacheAll = new HashMap<String, Video>();

    public YouTubeCacheProxy() {
        this.youtubeService = new ThirdPartyYouTubeClass();
    }

    @Override
    public HashMap<String, Video> popularVideos() {
        if (cachePopular.isEmpty()) {
            cachePopular = youtubeService.popularVideos();
        } else {
            System.out.println("Retrieved list from cache.");
        }
        return cachePopular;
    }

    @Override
    public Video getVideo(String videoId) {
        Video video = cacheAll.get(videoId);
        if (video == null) {
            video = youtubeService.getVideo(videoId);
            cacheAll.put(videoId, video);
        } else {
            System.out.println("Retrieved video '" + videoId + "' from cache.");
        }
        return video;
    }

    public void reset() {
        cachePopular.clear();
        cacheAll.clear();
    }
}
```

downloader

downloader/YouTubeDownloader.java: Media downloader app

```
import refactoring_guru.proxy.example.some_cool_media_library.ThirdPartyYouTubeLib;
import refactoring_guru.proxy.example.some_cool_media_library.Video;

import java.util.HashMap;

public class YouTubeDownloader {
    private ThirdPartyYouTubeLib api;

    public YouTubeDownloader(ThirdPartyYouTubeLib api) {
        this.api = api;
    }

    public void renderVideoPage(String videoId) {
        Video video = api.getVideo(videoId);
        System.out.println("\n-------------------------------");
        System.out.println("Video page (imagine fancy HTML)");
        System.out.println("ID: " + video.id);
        System.out.println("Title: " + video.title);
        System.out.println("Video: " + video.data);
        System.out.println("-------------------------------\n");
    }

    public void renderPopularVideos() {
        HashMap<String, Video> list = api.popularVideos();
        System.out.println("\n-------------------------------");
        System.out.println("Most popular videos on YouTube (imagine fancy HTML)");
        for (Video video : list.values()) {
            System.out.println("ID: " + video.id + " / Title: " + video.title);
        }
        System.out.println("-------------------------------\n");
    }
}
```

Demo.java: Initialization code

```
package refactoring_guru.proxy.example;

import refactoring_guru.proxy.example.downloader.YouTubeDownloader;
import refactoring_guru.proxy.example.proxy.YouTubeCacheProxy;
import refactoring_guru.proxy.example.some_cool_media_library.ThirdPartyYouTubeClass;

public class Demo {

    public static void main(String[] args) {
        YouTubeDownloader naiveDownloader = new YouTubeDownloader(new ThirdPartyYouTubeCl
        YouTubeDownloader smartDownloader = new YouTubeDownloader(new YouTubeCacheProxy()
```

```
        long smart = test(smartDownloader);
        System.out.print("Time saved by caching proxy: " + (naive - smart) + "ms");

    }

    private static long test(YouTubeDownloader downloader) {
        long startTime = System.currentTimeMillis();

        // User behavior in our app:
        downloader.renderPopularVideos();
        downloader.renderVideoPage("catzzzzzzzzz");
        downloader.renderPopularVideos();
        downloader.renderVideoPage("dancesvideoo");
        // Users might visit the same page quite often.
        downloader.renderVideoPage("catzzzzzzzzz");
        downloader.renderVideoPage("someothervid");

        long estimatedTime = System.currentTimeMillis() - startTime;
        System.out.print("Time elapsed: " + estimatedTime + "ms\n");
        return estimatedTime;
    }
}
```

OutputDemo.txt: Execution result

```
Connecting to http://www.youtube.com... Connected!
Downloading populars... Done!

--------------------------------
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
--------------------------------

Connecting to http://www.youtube.com/catzzzzzzzzz... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
--------------------------------
```

```
Downloading populars... Done!

--------------------------------
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
--------------------------------

Connecting to http://www.youtube.com/dancesvideoo... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: dancesvideoo
Title: Some video title
Video: Random video.
--------------------------------

Connecting to http://www.youtube.com/catzzzzzzzzz... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
--------------------------------

Connecting to http://www.youtube.com/someothervid... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: someothervid
Title: Some video title
Video: Random video.
--------------------------------

Time elapsed: 9354ms
Connecting to http://www.youtube.com... Connected!
Downloading populars... Done!

--------------------------------
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
```

```
--------------------------------

Connecting to http://www.youtube.com/catzzzzzzzzz... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
--------------------------------

Retrieved list from cache.

--------------------------------
Most popular videos on YouTube (imagine fancy HTML)
ID: sadgahasgdas / Title: Catzzzz.avi
ID: asdfas3ffasd / Title: Dancing video.mpq
ID: 3sdfgsd1j333 / Title: Programing lesson#1.avi
ID: mkafksangasj / Title: Dog play with ball.mp4
ID: dlsdk5jfslaf / Title: Barcelona vs RealM.mov
--------------------------------

Connecting to http://www.youtube.com/dancesvideoo... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: dancesvideoo
Title: Some video title
Video: Random video.
--------------------------------

Retrieved video 'catzzzzzzzzz' from cache.

--------------------------------
Video page (imagine fancy HTML)
ID: catzzzzzzzzz
Title: Some video title
Video: Random video.
--------------------------------

Connecting to http://www.youtube.com/someothervid... Connected!
Downloading video... Done!

--------------------------------
Video page (imagine fancy HTML)
ID: someothervid
Title: Some video title
Video: Random video.
```