

James Foster

ESUG (Lyon 2023)



Using GemStone/S for Web Applications

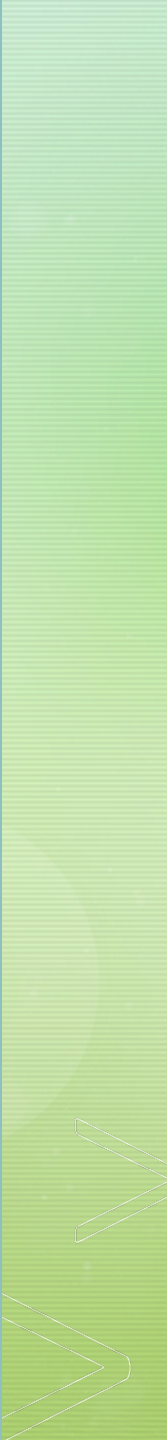



Disclaimer

- These are personal projects



Agenda

- Installing GemStone/S 64 Bit on Ubuntu with Debian package
 - WebGS as a back-end for modern web applications
- 



ESUG 2023 (Lyon)

Debian Package



Debian Package Manager

WIKIPEDIA The Free Encyclopedia

Search Wikipedia

Create account Log in

deb (file format)

26 languages

Contents [hide]

(Top)

Design

Implementation

- Control archive
- Signed packages

Adoption

See also

References

External links

Article Talk

From Wikipedia, the free encyclopedia

deb is the format, as well as [filename extension](#) of the [software package](#) format for the [Debian Linux distribution](#) and [its derivatives](#).


Design [edit]

Debian packages are standard [Unix ar archives](#) that include two [tar](#) archives. One archive holds the control information and another contains the installable data.^[2]

[dpkg](#) provides the basic functionality for installing and manipulating Debian packages. Generally end users don't manage packages directly with [dpkg](#) but instead use the [APT](#) package management software or other [APT front-ends](#) such as [aptitude](#) ([nCurses](#)) and [synaptic](#) ([GTK](#)).^[3]

Debian packages can be converted into other package formats and vice versa using [alien](#), and created from source code using [checkinstall](#) or the Debian Package Maker.^[4]

Debian package



The [GNOME](#) icon for deb files.
(Showing the [Debian](#) logo on a package)

Filename extension	.deb, .udeb
Internet media type	application/vnd.debian.binary-package ^[1]
Developed by	Debian
Type of format	Package management system
Container for	Software package
Extended from	ar archive , tarball
Website	deb format specification ↗



Using GemTalk's alpha-ppa

- Add the GemStone package repository to your repository list
- Add the GemTalk public key to your keyring
- Update the local package list
- Install GemStone

<https://alpha-ppa.gemtalksystems.com>

GemStone/S 64 Bit Debian Package Repository

To use this package repository, add it to your sources list with a public key, then update and install!

```
sudo curl -o /etc/apt/sources.list.d/gemstone.sources https://alpha-ppa.gemtalksystems.com/gemstone.sources
sudo curl -o /etc/apt/keyrings/gemstone.gpg https://alpha-ppa.gemtalksystems.com/gemstone.gpg
sudo apt update
sudo apt install gemstone-server
```

This will ask you some questions and do the install. To "preseed" answers for a non-interactive install, do the following:

```
echo gemstone-server gemstone-server/password password mySecret | debconf-set-selections
echo gemstone-server gemstone-server/password-confirm password mySecret | debconf-set-selections
echo gemstone-server gemstone-server/unicode-comparison boolean yes | debconf-set-selections
export DEBIAN_FRONTEND=noninteractive
sudo apt update -q
sudo apt install -q -y gemstone-server
```

Alternatively, you can [browse](#) the repository directly, download the appropriate package, and use `sudo dpkg --install gemstone-server_*.deb` to install it.

See [this video](#) for an example of how to install it.



3-minute Demo

- Create an Ubuntu 22.04.3 Server virtual machine
- Install GemStone
- Browse directories

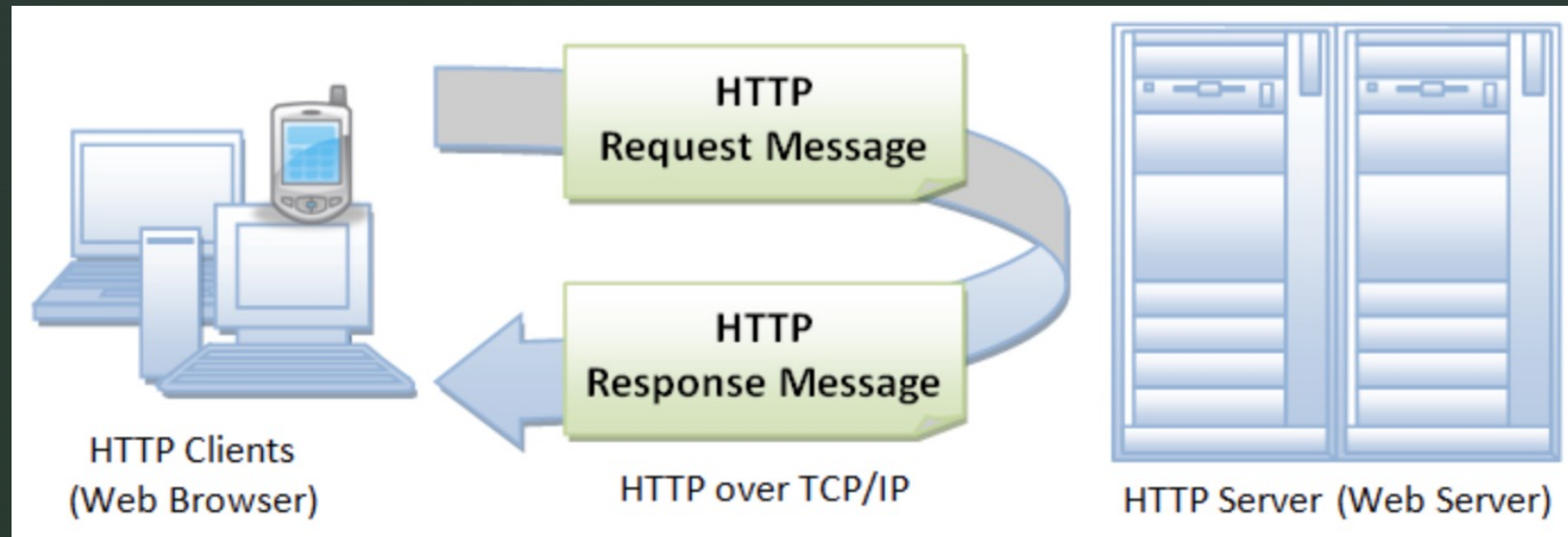


ESUG 2023 (Lyon)

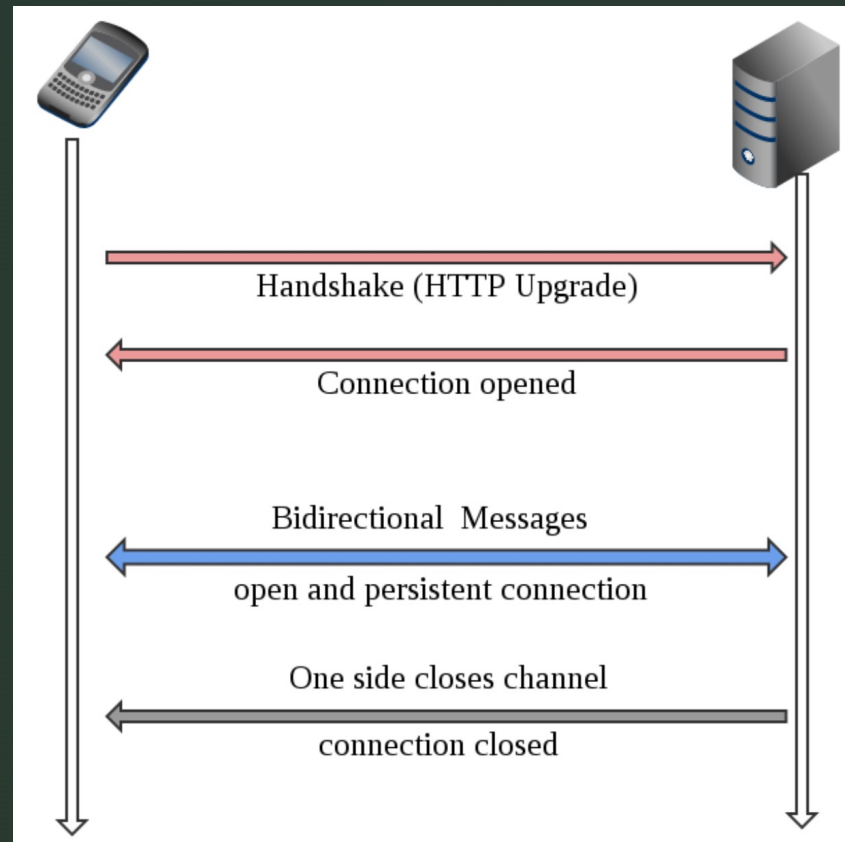
WebGS



Quick HTTP Primer



WebSockets





Demo with Implicit Routes

- Method `foo_gs` will be handler for `/foo.gs`

Routes and Handlers in Shelf (Dart)

```
import 'package:shelf_router/shelf_router.dart';
import 'package:shelf/shelf.dart';
import 'package:shelf/shelf_io.dart' as io;

var app = Router();

app.get('/hello', (Request request) {
  return Response.ok('hello-world');
});

app.get('/user/<user>', (Request request, String user) {
  return Response.ok('hello $user');
});

var server = await io.serve(app, 'localhost', 8080);
```

Routes in Express (Node.JS)

```
// GET method route
app.get('/', (req, res) => {
  res.send('GET request to the homepage')
})

// POST method route
app.post('/', (req, res) => {
  res.send('POST request to the homepage')
})

app.get('/users/:userId/books/:bookId', (req, res) => {
  res.send(req.params)
})
```

Explicit Routes in WebGS

```
| app |  
app := Router new  
  get: '/films' do: [:req :res |  
    res send: Film films.  
  ];  
  get: '/films/:id/title' do: [:req :res :id |  
    res send: (Film withId: id asNumber) title.  
  ];  
  get: '/films/:id/views' do: [:req :res :id |  
    res send: (Film withId: id asNumber) views.  
  ];  
  get: '/films/:id' do: [:req :res :id |  
    res send: (Film withId: id asNumber).  
  ];  
  post: '/films/:id/views/:n' do: [:req :res :id :n |
```



Demo of Explicit Routes



Code and Packages

- Code
 - <https://github.com/jgfoster/WebGS>
 - https://github.com/jgfoster/gemstone_deb
- Package Repository
 - <https://alpha-ppa.gemtalksystems.com>

Contact, Questions, and Credits

- Contact

- James.Foster@GemTalkSystems.com
- James.Foster@WallaWalla.edu
- <https://programminggems.wordpress.com/>

- Credits

- https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html
- https://www.researchgate.net/figure/Web-socket-architecture_fig3_338553959