

Web App Starter Kit

Welcome to this quick start guide

This page will give you **step-by-step instructions** on how to build **your first Web App from scratch**. The new APP will go in place of your existing Companion Site, that is will be visible on your workstation when visiting localhost:8088 (<http://localhost:8088>) (or whatever URL you assigned to it).

What you will need

This information refers only to DBFree V4.1 (it may also apply to V4.0 but it is not guaranteed to work).

You will need to **have DBFree installed** and working fine and to keep the Code Assistant Service page open. Nothing else.

Step 1 - preparing DBFree

From the Control Center

1. Click on **Open web root**
 2. **Navigate** to its upper folder (i.e. c:\dbfreev41)
 3. **Rename** your actual WEB folder to something else (i.e. WEB1)
 4. **Create** a new folder and name it WEB (like the previous one)
-

Step 2 - getting the framework

Download, from this section of the Assistant, the following compressed files:

1. **css.zip** : contains the folder with all the CSS you will use in your web app.
2. **js.zip** : contains the folder with jQuery, Bootstrap and jMax javascripts.
3. **common.zip** : contains the basic files of the DBFree Application Framework.

You can store these files somewhere or use them immediately opening them from browser. We suggest to store it for later re-use.

Step 3 - preparing the ground

Now you should uncompress the zip files and extract the folder they contain, so to create this tree structure:

```
c:\--- DBFreeV41 --- (file:///c:/---DBFreeV41---)           +-- common
    |           |
    +-- WEB ---+-- css
        |
        +-- js
```

Check to have correctly unzipped all files of each folder.

The basic framework (a very basic one) for your new application is ready to accept the web pages.

Step 4 - Adding web pages

Download, always from this same section, the following compressed files:

1. **webroot_files.zip** : contains the **WEB entrypoint file** and **some other** to populate your app.
2. **images_kit.zip** - contains the **image** folder with some icons and images that can be useful.

Repeate the process of uncompressing so to have this final structure:

```
c:\--- DBFreeV41 --- (file:///c:/---DBFreeV41---)
    +-- WEB ----- css
        +-- js
        +-- images
        +
        home.msp (ecc)
```

Your new Web App is now ready for customization.

Customizing

By far, the simplest way of customizing your application is to **add first the static contents** and only then, once you are **satisfied of the look** of your pages, proceed with **developing the code** to provide live data.

Editing pages is simple: it can be done by **double-clicking** on the file (i.e. the home page we have here in our new site): if you have installed DBFree correctly you should see the page opening into **PSPad editor**.

If this does not happens you should set the files with **.msp** extension to be always opened by using PSPad using Windows own procedure.

Of course a web application is seldom implemented by using a single page (even if is not so uncommon as it may seems), so sooner or later you will need to add some toher pages. This is easy.

Step 5 - Adding new web pages

The standard procedure to add pages is this:

1. **right-click** on the **WEB** folder
2. from context menu **choose** New Text Document
3. **change** the default name to desired one and the extension from **.txt** to **.msp**
4. Go to Code Assistant and **choose a page template** that fits for the intended use of the page (i.e. 4017-STD page template)
5. **copy** from the template and **paste** into your newly created page
6. save the page and start editing (i.e. put a link to go from this new page back to home page)

Don't forget to go to your home page and **add a link to the new page**.

Where to get help with HTML

HTML5 is reach of options and opens lot of possibilities. Too many to memorize them all.

We **suggest you to keep a window open on this web site (<http://w3schools.com>)** for getting a real help on HTML (and Javascript).

From a simple web to a Web App

If you followed the instructions so far you should have **a web fully working**, made out of dynamic pages, but that does nothing more than a static, classic web site. You spent far more less time than building it *piece-by-piece by hand* but substantially it seems to do the same job.

In facts there is a difference: if you click on `List of files of this folder` you will get a **self-updating list**. It means you **don't have to update manually** each and every link **everytime you add a page** or change its name.

It's not bad, but of course we expect much more from our efforts and to justify the hassle to install a language for the web. Well, this is not even a foretaste of what DBFre can do.

Adding features

To really see the difference you **should add some other components** to your web:

Download, always from this same section, the compressed file:**app_folder_kit.zip**.

Do the usual process of **uncompressing** so to have this final structure:

```
+-- APPS ----- (subfolders)
|
c:\--- DBFreeV41 ---+ (file:///c:/---DBFreeV41---+)
    +-- WEB --+-- css
        +-- js
        +-- images
    +
    home.msp (etc)
```

Now **refresh your home page**: if you did everything well you should see that a new button appeared: `List availabel DBApps`.

Building DBApps

Still from Windows, **double click** on the newly added APP folder and **open it**:

1. **Right-click** and from context menu **choose New Folder**
2. **Name** your newly created folder something like **MyFirstApp** (or anything else, but avoid blank spaces)
3. **Open** the new folder and **create**, with same system, **a new text file** then name it `start.msp`

The final structure should be something like this:

```
+-- APPS -----+-- list_apps.msp
|
c:\--- DBFreeV41 ---+ (file:///c:/---DBFreeV41---+)
    +-- common
    +-- MyFirst
    App ---+-- start.msp
        +-- WEB --+-- css
            +-- js
            +-- images
        +
        home.msp (etc)
```

Now if you **click** on `List availabel DBApps` in your Home Page **you'll see a menu** and your **new DBApp** listed.

Now, by beginning editing this new **start page** you are ready for developing. This means **adding** the correct **header**, the **code** and all the **program logic** you'll need until you'll have the application doing what you plan to make it do.

But starting from a blank page is not the easiest way, isn't it?

Step 6 - Adding a pre-built DBApp

Having **something to start from** may really be of great help.

1. **Download** the compressed file:**Demo_DBApp.zip**.
2. The file contains a folder: **uncompress** it into your APPS folder
3. Come back to **click** on List availabel DBApps
4. **Click** on the button **to launch** the application

```

          +-- APPS -----+--list_apps.msp
          |
c:\--- DBFreeV41 --- (file:///c:/---DBFreeV41---)      +-- common           +-- yourApp
1 ---start.msp (etc)
          +-- WEB ---+-- css           +-- yourApp2 ---start.msp
          +-- js            +-- (etc)
          +-- images
          +
          home.msp (etc)

```

Now you have a working demo application to study and use as inspiration. And eventually start modifying to see what happens.

How it works

Before to start writing your own code it's better to have a **general understanding** of the organization of the hosting environment.

Home page

The **home page** of your web application is responsible of **initializing the whole thing** by mean of the **MAX Header** that sits on top.

The **header** does the followings:

1. **loads** the necessary library (there are many of them, for special purposes and different versions of DBX family of products: in this case we use a generic one ideal for DBFree)
2. **initializes** the **global variables** that holds the values that apply to any page and for all users (the common parameters for all the application, like folders to use and the like)
3. **initializes** (or recall) the **user's sessions** that keeps isolated the variables of the same names that you use in the code from one user to another.
4. sets the database driver to use (in this case the proprietary MaxScript engine): as you may suspect MaxScript can work with different databases, but DBFree usually supports only the proprietary one. With this driver you can use Clipper and FoxPro indexes. DBase and ODBC are only supported using optional drivers.
5. **check** if the **folders** necessary to the application exists and in case creates them
6. **check** for incoming **parameters** (WebVars) and **stores their values into memory** (local) variables to be used with MaxScript

Session handling

Session handling would require an entire chapter to be explained: fortunately **you don't need to know all details** (at least not straight away) and all you need to know is that for **preserving** the values of **variables** you will have to **use the seed**.

What is a **seed**?

The **seed** is only a **number**, generated by the server, **that identifies univocally** a given user and its connection. This number is **assigned automatically** when the **users** connect (via their browsers) the first time and then **is stored into their browser**.

DBFree V41 **uses** a feature of modern web **browsers** **HTML5 compliant** (the vast majority) called **local storage** to accomplish this task.

This feature **consents to store** arbitrary values into **browser's storage system** without using a cookie and is by far the most efficient method to identify an user that connects from internet.

How do you will use seeds?

You should **pass the seed to any page** below your home page.

There are two basic methods to do that:

- using a **web form field**: <input type="text" name="XY" value="<%>cSeed%>">
- using a **QueryString**: <a href="nextpage.msp?XY=<%>cSeed%>">Go to Next Page

We are not going to discuss this topic here: just **keep in mind** that even if **you don't pass the seed** the code in the DBFree headers **will try to recover it automatically**, so - in theory - you could omit to pass the seed from one page to another and **rely to the automatisms**.

Anyway, considered the many browsers and versions crawling around the web, it's better to use seeds whenever possible (or adopt more advanced techniques not discussed here).

[Top of page](#)

[Open this page in a full window](#)