



Join the Stack Overflow Community

Stack Overflow is a community of 6.3 million programmers, just like you, helping each other. Join them; it only takes a minute:

Sign up

Transport security has blocked a cleartext HTTP



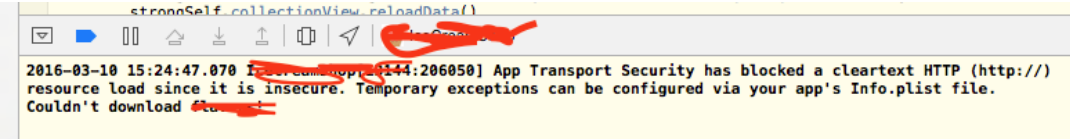
Got 5 minutes?

Then you have time to deploy your first cloud solution.

Try Azure — free

What setting do I need to put in my `info.plist` to enable `http` mode as per the following error message?

Transport security has blocked a cleartext HTTP (`http://`) resource load since it is insecure. Temporary exceptions can be configured via your app's `Info.plist` file.



[ios](#) [xcode](#) [ios9](#) [ios10](#) [app-transport-security](#)

edited Oct 24 at 8:46

asked Jul 6 '15 at 20:10



Jeef

7,544 7 26 75

1 [stackoverflow.com/a/31629980/1803879](#) – Tom Howard Sep 1 '15 at 14:48

1 **Moderator Note:** There are already 36 answers to this question. Before adding *another*, please make sure your solution is **new**. – Matt ♦ Jul 11 at 17:45

See this tutorial why required : [iosdevcenters.blogspot.in/2016/02/...](#) – Kirit Modi Jul 31 at 5:22

25 Answers

Here are the settings visually:



answered Nov 14 '15 at 19:08



4.670 1 5 7

3 This option does not exist - XCode 7.3.1 – [jameshfisher](#) Jun 23 at 13:14



NSAppTransportSecurity	Dictionary	(1 item)
NSAllowsArbitraryLoads	Boolean	YES

answered Aug 4 '15 at 10:42



	8.205	3	7	14
CristiCh				

9 While this solution is known to be vulnerable, it is the **only** solution I would recommend during **DEVELOPMENT**. Having to type in every exact domain during development is just silly (especially if you

are using third party web services). – [reTs](#) Nov 3 '15 at 11:42

- 68 LET IT BE KNOWN @Jackson: this is not a "workaround" nor a "hack": it is fixing Apple's absurd bug. Forcing all HTTP traffic to go over HTTPS is *nothing to do with security*. If you think that makes your traffic secure, you have a lot of reading + re-education to do. This is a political move by Apple, with some benefits, but also with down-sides. It is nothing to do with "lazy developers" or similar. – [Adam](#) Nov 5 '15 at 0:05
- 5 It is not at all obvious how ATS can improve security: as a security professional, I promise you it DOES NOT improve security. It has no effect. There are obvious ways that ATS breaks some of Apple's competitors' business models (coughGooglecough), but otherwise it is a very strange thing to do. – [Adam](#) Dec 11 '15 at 23:18

See the forum post [Application Transport Security?](#).

Also the page [Configuring App Transport Security Exceptions in iOS 9 and OSX 10.11](#).

For example, you can add a specific domain like:

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>yourserver.com</key>
    <dict>
      <!--Include to allow subdomains-->
      <key>NSIncludesSubdomains</key>
      <true/>
      <!--Include to allow HTTP requests-->
      <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
      <true/>
      <!--Include to specify minimum TLS version-->
      <key>NSTemporaryExceptionMinimumTLSVersion</key>
      <string>TLSv1.1</string>
    </dict>
  </dict>
</dict>
```

The lazy option is:

```
<key>NSAppTransportSecurity</key>
<dict>
  <!--Include to allow all connections (DANGER)-->
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

Note:

info.plist is an XML file so you can place this code more or less anywhere inside the file.

edited Jul 2 at 19:03



[Peter Mortensen](#)

10.3k 13 69 107

answered Jul 6 '15 at 20:19



[KMLong](#)

6,045 2 7 12

- 14 not working for me with iOS9 beta 3. Any solution? – [mursang](#) Jul 11 '15 at 13:34
- 2 Doesn't seem to be working in Beta 4 either. – [user2443329](#) Jul 24 '15 at 20:46
- 2 @Imiguelvargasf open your info.plist in a plain text editor – [Dan Beaulieu](#) Jul 30 '15 at 20:24
- 3 I still get the error: exception domain is set and NSAllowsArbitraryLoads is false. Even with NSAllowsArbitraryLoads set true the error shows up. Does anyone else here have this issue? – [very_supercharged](#) Oct 13 '15 at 12:57
- 2 As of 1/30/2016 Apple doc shows that the keys no longer have the Temporary word in them, e.g.: NSExceptionAllowsInsecureHTTPLoads NSExceptionMinimumTLSVersion See [developer.apple.com/library/prerelease/ios/documentation/...](#) – [Philippe Monnet](#) Jan 30 at 21:32

This was tested and was working on iOS 9 GM seed - this is the configuration to allow a *specific* domain to use HTTP instead of HTTPS:

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <false/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>yourdomain.com</key>
    <dict>
      <key>NSIncludesSubdomains</key>
```

```

<true/>
<key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
<true/>
<key>NSTemporaryExceptionMinimumTLSVersion</key>
<string>TLSv1.1</string>
</dict>
</dict>
</dict>

```

NSAllowsArbitraryLoads must be false, because it disallows **all** insecure connection, but the exceptions list allows connection to **some** domains without HTTPS.

edited Jul 2 at 19:35



Peter Mortensen

10.3k 13 69 107

answered Sep 14 '15 at 8:36



Sound Blaster

2,677 1 12 25

5 THIS should be marked as the answer. Tested and working on iOS 9 GM seed to allow a specific domain to use http without taking the "lazy" way and completely opening your app up. – [DiscDev](#) Sep 16 '15 at 17:18

2 How do I add this to my info.plist? – [JMStudios.jrichardson](#) Sep 20 '15 at 1:33

2 This is finally the correct answer to the question. Thank you. – [dpizzuto](#) Sep 20 '15 at 9:38

2 Ok I added that entry to my info.plist and I'm still getting this error - " App Transport Security has blocked a cleartext HTTP (http://) resource load since it is insecure. Temporary exceptions can be configured via your app's Info.plist file." – [user30646](#) Oct 5 '15 at 16:57

1 @RomanShapovalov Using IP addresses is strongly not recommended because all iOS apps must conform IPv6, see Apple Docs about this – [Sound Blaster](#) Sep 27 at 9:39

This is a quick workaround (but not recommended) to add this in the plist:

```

<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>

```

Which means (according to [Apple's documentation](#)):

NSAllowsArbitraryLoads

A Boolean value used to disable App Transport Security for any domains not listed in the NSExceptionDomains dictionary. Listed domains use the settings specified for that domain.

The default value of NO requires the default App Transport Security behaviour for all connections.

I really recommend links:

- [Apple's technical note](#)
- [WWDC 2015 session 706 \(Security and Your Apps\)](#) starts around 1:50
- [WWDC 2015 session 711 \(Networking with NSURLSession\)](#)
- Blog post [Shipping an App With App Transport Security](#)

which help me understand reasons and all the implications.

The XML (in file Info.plist) below will:

```

<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <false/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>PAGE_FOR_WHICH_SETTINGS_YOU_WANT_TO_OVERRIDE</key>
    <dict>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
    </dict>
  </dict>
</dict>

```

disallow arbitrary calls for all pages, but for PAGE_FOR_WHICH_SETTINGS_YOU_WANT_TO_OVERRIDE will allow that connections use the HTTP protocol.

To the XML above you can add:

```

<key>NSIncludesSubdomains</key>
<true/>

```

if you want to allow insecure connections for the subdomains of the specified address.

The best approach is to block all arbitrary loads (set to false) and add exceptions to allow only addresses we know are fine.

[For interested readers](#)

edited Jul 3 at 10:42

answered Sep 1 '15 at 12:10



[Julian Król](#)

5,847 4 23 51

1 NSAllowsArbitraryLoads must be false – [Sound Blaster](#) Sep 14 '15 at 8:37

@SoundBlaster for what case and what you see wrong in my answer to down vote? – [Julian Król](#) Sep 21 '15 at 9:20

by adding in plist(NSAppTransportSecurity NSAllowsArbitraryLoads), all web service working fine except one web service one web service return internal server error (500) in iOS 9 but working fine in iOS8 or later – [amit gupta](#) Sep 22 '15 at 10:19

@SoundBlaster made a change, now you shouldn't have objections :) – [Julian Król](#) Oct 7 '15 at 7:36

2 Great explanation, great links provided. This is a wise solution. – [LargeGlasses](#) Nov 20 '15 at 15:26

For those of you who want a more context on **why** this is happening, in addition to how to fix it, then read below.

With the introduction of iOS 9, to improve the security of connections between an app and web services, *secure connections between an app and its web service must follow best practices*.

The best practices behavior is enforced by the **App Transport Security** to:

- prevent accidental disclosure, and
- provide a default behavior that is secure.

As explained in the [App Transport Security Technote](#), when communicating with your web service, App Transport Security now has the following requirements and behavior:

- The server must support at least Transport Layer Security (TLS) protocol version 1.2.
- Connection ciphers are limited to those that provide forward secrecy (see the list of ciphers below.)
- Certificates must be signed using a SHA256 or better signature hash algorithm, with either a 2048 bit or greater RSA key or a 256 bit or greater Elliptic-Curve (ECC) key.
- Invalid certificates result in a hard failure and no connection.

In other words, your web service request should: a.) use **HTTPS** and b.) be encrypted using TLS v1.2 with forward secrecy.

However, as was mentioned in other posts, you can override this new behavior from App Transport Security by specifying the insecure domain in the `Info.plist` of your app.

To override, you will need to add the `NSAppTransportSecurity > NSExceptionDomains` dictionary properties to your `Info.plist`. Next, you will add your web service's domain to the `NSExceptionDomains` dictionary.

For example, if I want to bypass the App Transport Security behavior for a web service on the host **www.yourwebservicehost.com** then I would do the following:

1. Open your app in Xcode.
2. Find the `Info.plist` file in Project Navigator and "right-mouse" click on it and choose the `Open As > Source Code` menu option. The property list file will appear in the right pane.
3. Put the following properties block inside of the main properties dictionary (under the first `<dict>`).

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>www.yourwebservedomain.com</key>
    <dict>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
      <key>NSExceptionMinimumTLSVersion</key>
      <string>TLSv1.1</string>
```


```
<key>NSIncludesSubdomains</key>
<true/>
</dict>
</dict>
</dict>
```

If you need to provide exceptions for additional domains then you would add another dictionary property beneath `NSExceptionDomains`.

To find out more about the keys referenced above, read [this already mentioned technote](#).

edited Sep 21 '15 at 22:18

answered Sep 21 '15 at 17:56



whyceewhite

3,062 4 21 38

1

This worked in 2 of my apps, but it's not working on a 3rd one. Has anyone else encountered a situation of using the above fix and still getting the same error message? (and yes I updated the dictionary to use my API domain, not the one in the code) – [helloB](#) Sep 22 '15 at 21:41

Best ! Apple confirms that this solution using – [YannickSteph](#) Oct 6 '15 at 8:01

Best answer, working solution. – [Dustin Williams](#) Oct 7 '15 at 13:30

This worked for a Cordova/Phonegap/Ionic app editing file `./platforms/ios/<project>/<project>-Info.plist` with `NSAllowsArbitraryLoads=false` and many exception domains to services with varying TLS/HTTP/HTTPS combinations. Initially used `NSAllowsArbitraryLoads=true` then adjusted, troubleshooting the rules through trial and error to comply with guidelines and submit for approval. Note `config.xml` `<access origin=.../>` statements partially populate this file, but currently requires adjustment by direct editing or via XCode to get the details correct. – [jimmont](#) Dec 10 '15 at 20:10

Also `<access origin="*" />` (in `config.xml`) sets `NSAllowsArbitraryLoads=true` (for Cordova/Phonegap/hybrid apps – [jimmont](#) Dec 10 '15 at 20:19


I do not like editing the plist directly. You can easily add it to the plist using the GUI:

- Click on the Info.plist in the Navigator on the left.
- Now change the data in the main area:
 - On the last line add the +
 - Enter the name of the group: **App Transport Security Settings**
 - Right click on the group and select `Add Row`
 - Enter **Allow Arbitrary Loads**
 - Set the value on the right to **YES**

Bundle creator	OS Type code	String	????
▼ App Transport Security Settings		Dictionary	(1 item)
Allow Arbitrary Loads		Boolean	YES
Bundle version		String	1


edited Jan 17 at 19:37

answered Nov 6 '15 at 8:06



gondo

449 5 18



Vincent

2,362 20 26

1

thank you. you saved my time. – [beginners](#) Nov 13 '15 at 3:01

that's why we are all doing this! – [Vincent](#) Nov 19 '15 at 15:06

To clarify: if "Allow Arbitrary Loads" is YES, and there are "Exception Domains", then the loads that are allowed are restricted to the ones in the Exception Domains. Is that correct? – [user3821934](#) Nov 20 '15 at 0:40

is that a safe way to do it if I want to release the application? – [Lamar](#) Feb 19 at 11:13

This not working for me why? – [mzonorz](#) Aug 19 at 10:06

Transport security is available on iOS 9.0 or later. You may have this warning when trying to call a WS inside your application:

Application Transport Security has blocked a cleartext HTTP (http://) resource load since it is insecure. Temporary exceptions can be configured via your app's Info.plist file.

Adding the following to your Info.plist will disable ATS:

```
<key>NSAppTransportSecurity</key>
<dict>
```

```
<key>NSAllowsArbitraryLoads</key><true/>
</dict>
```

edited Jul 2 at 19:09

answered Sep 18 '15 at 17:36



Peter Mortensen

10.3k 13 69 107



Malek Belkahla

363 2 14

- 1 NSAllowsArbitraryLoads must be false, if true it allows all insecure connection – [Thiago Arreguy](#) Sep 19 '15 at 12:03

by adding in plist all web service working fine except one web service one web service return internal server error (500) in iOS 9 but working fine in iOS8 or later – [amit gupta](#) Sep 22 '15 at 10:18

- 1 Accepted on the store? – [Slartibartfast](#) Sep 23 '15 at 9:15

App Transport Security

App Transport Security (ATS) enforces best practices in the secure connections between an app and its back end. ATS prevents accidental disclosure, provides secure default behavior, and is easy to adopt; it is also on by default in iOS 9 and OS X v10.11. You should adopt ATS as soon as possible, regardless of whether you're creating a new app or updating an existing one.

If you're developing a new app, you should use HTTPS exclusively. If you have an existing app, you should use HTTPS as much as you can right now, and create a plan for migrating the rest of your app as soon as possible. In addition, your communication through higher-level APIs needs to be encrypted using TLS version 1.2 with forward secrecy. If you try to make a connection that doesn't follow this requirement, an error is thrown. If your app needs to make a request to an insecure domain, you have to specify this domain in your app's Info.plist file.

The following listing represents the overall structure of the `NSAppTransportSecurity` dictionary, showing all possible keys (all of which are optional). Keep this structure in mind as you configure each element of the `NSAppTransportSecurity` dictionary.

```
NSAppTransportSecurity : Dictionary {
    NSExceptionDomains : Dictionary {
        <domain-name-string> : Dictionary {
            NSIncludesSubdomains : Boolean
            // Keys to describe your app's intended network behavior for
            // domains whose security attributes you control
            NSExceptionAllowsInsecureHTTPLoads : Boolean
            NSExceptionRequiresForwardSecrecy : Boolean
            NSExceptionMinimumTLSVersion : String
            // Keys to describe your app's intended network behavior for
            // domains whose security attributes you don't control
            NSThirdPartyExceptionAllowsInsecureHTTPLoads : Boolean
            NSThirdPartyExceptionRequiresForwardSecrecy : Boolean
            NSThirdPartyExceptionMinimumTLSVersion : String
        }
    }
    NSAllowsArbitraryLoads : Boolean
}
```

There are two solutions for this :

Solutions 1 :

1. In Info.plist file add a dictionary with key 'NSAppTransportSecurity '
2. Add another element inside dictionary with key 'Allow Arbitrary Loads'

Plist structure should appear as shown in below image.

Bundle version	String	1
▼ App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES

Solution 2 :

1. In Info.plist file add a dictionary with key 'NSAppTransportSecurity '
2. Add another element inside dictionary with key 'NSExceptionDomains '
3. Add element with key 'MyDomainName.com' of type NSDictionary
4. Add element with key 'NSIncludesSubdomains' of type Boolean and value set as YES
5. Add element with key 'NSTemporaryExceptionAllowsInsecureHTTPLoads' of type Boolean and value set as YES


Plist structure should appear as shown in below image.

Bundle version	String	1
▼ App Transport Security Settings	Dictionary	(1 item)
▼ Exception Domains	Dictionary	(1 item)
NSIncludesSubdomains	Boolean	YES
NSTemporaryExceptionAllowsInsecureHTTPLoads	Boolean	YES

Solution 2 is preferred since it allows only selected domain whereas solution 1 allows all insecure HTTP connections.

edited Mar 10 at 12:51

answered Mar 10 at 11:29


 [Jayprakash Dubey](#)
9,411 2 64 73

It works for me too; just copy and paste the following into your info.plist file:

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <false/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>yourdomain.com</key>
    <dict>
      <key>NSIncludesSubdomains</key>
      <true/>
      <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
      <true/>
      <key>NSTemporaryExceptionMinimumTLSVersion</key>
      <string>TLSv1.1</string>
    </dict>
  </dict>
</dict>
```

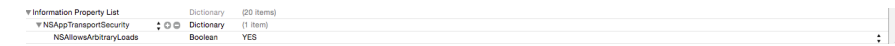
edited Jul 4 at 14:27

answered Sep 17 '15 at 2:41

 [Thiago Arreguy](#)
1,229 1 6 12

- 7 Does not add anything beyond what is in other answers already posted. – Bryan Sep 28 '15 at 20:57
- This killed my app :(– Agent Zebra Aug 10 at 19:12


Use:




Add a new item, **NSAppTransportSecurity**, in the plist file with type **Dictionary**, then add sub item **NSAllowsArbitraryLoads** in dictionary of type **Boolean**, and set bool value **YES**. This works for me.

edited Jul 2 at 19:10

answered Sep 21 '15 at 9:11

 [Peter Mortensen](#)
10.3k 13 69 107

 [Tejinder](#)
325 3 9

- 1 NSAllowsArbitraryLoads must be false, if true it allows all insecure connection – Thiago Arreguy Sep 21 '15 at 16:12
- by adding in plist all web service working fine except one web service one web service return **internal server error (500)** in iOS 9 but working fine in iOS8 or later @ThiagoArreguy – amit gupta Sep 22 '15 at 10:18


Development Example


Here is a screenshot of a plist which keeps ATS intact (=secure), but allows that connections to **localhost** can be made via **HTTP instead of HTTPS**. It works in Xcode 7.1.1.

▶ Required device capabilities	▼	Array	(1 item)
▶ Supported interface orientations	▼	Array	(3 items)
▼ App Transport Security Settings	⬆ ⬇ ⬇ ⬆	Dictionary	(1 item)
▼ Exception Domains	▼	Dictionary	(1 item)
▼ localhost	▼	Dictionary	(2 items)
NSIncludesSubdomains		Boolean	YES
NSExceptionAllowsInsecureHTTPLoads		Boolean	YES

edited Jul 2 at 19:23

answered Nov 30 '15 at 8:49

 [Peter Mortensen](#)
10.3k 13 69 107

 [Sebastian](#)
2,602 1 19 20

On 2015-09-25 (after Xcode updates on 2015-09-18):

I used a non-lazy method, but it didn't work. The followings are my tries.

First,

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>www.xxx.yyy.zzz</key>
    <dict>
      <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
      <true/>
      <key>NSTemporaryExceptionMinimumTLSVersion</key>
      <string>TLSv1.1</string>
      <key>NSIncludesSubdomains</key>
      <true/>
    </dict>
  </dict>
</dict>
```

And second,

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>www.xxx.yyy.zzz</key>
    <dict>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
      <key>NSExceptionMinimumTLSVersion</key>
      <string>TLSv1.1</string>
      <key>NSIncludesSubdomains</key>
      <true/>
    </dict>
  </dict>
</dict>
```

Finally, I used the lazy method:

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

It might be a little insecure, but I couldn't find other solutions.

edited Jul 2 at 19:30



Peter Mortensen

10.3k 13 69 107

answered Sep 25 '15 at 2:41



Heedoo

90 1 9

Figuring out what settings to use can be performed automatically, as mentioned in [this technote](#):

```
/usr/bin/nscurl --ats-diagnostics --verbose https://your-domain.com
```

answered Oct 7 '15 at 8:29



ecotax

1,628 12 19

You do need OS X 10.11 (El Capitan) for this, apparently. – [ecotax](#) Oct 7 '15 at 8:43

Update for Xcode 7.1, facing problem 27.10.15:

The new value in the Info.plist is "App Transport Security Settings". From there, this dictionary should contain:

- Allow Arbitrary Loads = YES
- Exception Domains (insert here your http domain)

edited Jul 2 at 19:25



Peter Mortensen

10.3k 13 69 107

answered Oct 27 '15 at 19:21



MkaysWork

152 2 8

It may be worth mentioning how to get there...

Info.plist is one of the files below the Main.storyboard or viewController.swift.

When you click on it the first time, it usually is in a table format, so right click the file and 'open as' Source code and then add the code below towards the end, i.e.:

```
<key>NSAppTransportSecurity</key><dict><key>NSAllowsArbitraryLoads</key><true/>
</dict>
```

Copy paste the code just above

```
"</dict>
</plist>"
```

which is at the end.

edited Jul 2 at 19:27



Peter Mortensen

10.3k 13 69 107

answered Oct 25 '15 at 17:18



Naishita

1,116 8 13

In Xcode 7 we need to do like this:

In file Info.plist, add:

- NSAppTransportSecurity As Dictionary & child as
- NSAllowsArbitraryLoads As Boolean value to be YES

edited Jul 2 at 19:33



Peter Mortensen

10.3k 13 69 107

answered Sep 22 '15 at 10:25



Uma Madhavi

2,882 5 11 30

hi, can u plz tel me how and where in info plist we have to add...i tried lots but my Xcode freeze.... i have my question not at answer stackoverflow.com/questions/32787595/... – user2813740 Sep 25 '15 at 18:31

at the end u can add.... <key>UISupportedInterfaceOrientations</key> <array>
<string>UIInterfaceOrientationPortrait</string> <string>UIInterfaceOrientationLandscapeLeft</string>
<string>UIInterfaceOrientationLandscapeRight</string> </array> <key>NSAppTransportSecurity</key>
<dict> <key>NSAllowsArbitraryLoads</key> <true/> </dict> </dict> </plist> – Uma Madhavi Sep 28 '15 at 7:34

Add this code into the info.plist file:

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key><true/>
</dict>
```

edited Jul 2 at 19:31



Peter Mortensen

10.3k 13 69 107

answered Sep 24 '15 at 4:14



Jayesh Miruliya

968 3 8

3 Does not add anything beyond what is in other answers already posted. – Bryan Sep 28 '15 at 20:59

Just use **Info.plist** inside easy and effective way;

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key><true/>
</dict>
```

answered Aug 24 at 8:59



SwiftDeveloper

589 1 5 26

According to Apple, generally disabling ATS will lead to app rejection, unless you have a good reason to do so. Even then, you should add exceptions for domains that you can access

safely.

Apple has an excellent tool that tells you exactly what settings to use: In Terminal, enter

```
/usr/bin/nscurl --ats-diagnostics --verbose https://www.myurl.com/whatever
```

and nsurl will check whether this request fails, and then try a variety of settings and tell you exactly which one passes, and what to do. For example, for some third-party URL that I visit, this command told me that this dictionary passes:

```
{
  NSExceptionDomains = {
    "www.thirdparty.com" = {
      NSExceptionRequiresForwardSecrecy = false;
    };
  };
}
```

To distinguish between your own sites and third-party sites that are out of your control, use, for example, the key `NSThirdPartyExceptionRequiresForwardSecrecy`.

edited Jul 2 at 19:22



Peter Mortensen

10.3k 13 69 107

answered Apr 17 at 23:54



gnasher729

32.4k 2 29 53

NOTE: The exception domain in your plist should be in LOWER-CASE.

Example: you have named your machine "MyAwesomeMacbook" under Settings->Sharing; your server (for test purposes) is running on MyAwesomeMacbook.local:3000, and your app needs to send a request to <http://MyAwesomeMacbook.local:3000/files...>, your plist you will need to specify "myawesomemacbook.local" as the exception domain.

--

Your info.plist would contain...

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>myawesomemacbook.local</key>
    <dict>
      <!--Include to allow subdomains-->
      <key>NSIncludesSubdomains</key>
      <true/>
      <!--Include to allow HTTP requests-->
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
    </dict>
  </dict>
</dict>
```

answered Sep 12 at 18:17



ObjectiveTC

487 6 10

For Cordova, if you want to add it into your ios.json, do the following:

```
"NSAppTransportSecurity": [
  {
    "xml": "<dict><key>NSAllowsArbitraryLoads</key><true /></dict>"
  }
]
```

And it should be inside of:

```
"*-Info.plist": {
  "parents": {
  }
}
```

answered Mar 21 at 12:41



zeusstl

577 6 13

Just `<access origin="" />` in config.xml didn't work. Not sure if both are necessary, but the ios.json solution sealed the deal for me. – zeusstl Mar 21 at 12:45

If you are using Xcode 8.0 and swift 3.0 or swift 2.2.

Application requires iPhone enviro...	Boolean	YES
▼ App Transport Security Settin...	Dictionary	(2 items)
Allow Arbitrary Loads	Boolean	YES
▼ Exception Domains	Dictionary	(1 item)
▼ domain.com	Dictionary	(2 items)
NSExceptionAllowsInsecureH...	Boolean	YES
NSIncludesSubdomains	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main

edited Oct 28 at 7:24

answered Oct 28 at 7:06

 **Anit Kumar**
625 4 5

For me, adding **NSAppTransportSecurity** and **NSExceptionDomains** directly into the info.plist file **via the standard Xcode Editor did not work**. (Adding it to the file in a textedit in the style mentioned above did work fine.)


After adding it via the textedit, info.plist looks like this:


▼ App Transport Security Settin...	Dictionary	(1 item)
▼ Exception Domains	Dictionary	(2 items)
graph.facebook.com	Dictionary	(2 items)

Therefore, I think you have to type these *words* into info.plist if you do it directly by Xcode. These words ("App Transport ..." and "Exception Do ..") are also suggested by Xcode if you begin to type them into the textfield.

edited Jul 2 at 19:24

answered Nov 3 '15 at 18:53

 **Peter Mortensen**
10.3k 13 69 107

 **Burn**
170 3 16

Like many have noted, this is a feature issue that comes with iOS 9.0. They have added a thing called App Transport Security, and I too was annoyed when it broke my Apps.


You can bandage it with the NSAllowsArbitraryLoads key to YES under NSAppTransportSecurity dictionary in your .plist file, but ultimately you will need to re-write the code that forms your URLs to form the HTTPS:// prefix.


Apple has re-written the NSURLConnection class in iOS 9.0. You can read about it in [NSURLConnection](#).

Else, you may have to back out of iOS 9.0 until you have time to implement the correct solution.

edited Jul 2 at 19:28

answered Oct 1 '15 at 18:41

 **Peter Mortensen**
10.3k 13 69 107

 **Mark DeWald**
1 1

protected by **Yogesh Suthar** Jan 6 at 13:46

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus](#) does not count).

Would you like to answer one of these [unanswered questions](#) instead?