AppGoatを利用した集合教育補助資料 -Fiddlerの使い方-

独立行政法人情報処理推進機構（IPA）
セキュリティセンター

Fiddlerとは

- HTTP/HTTPS リクエストやレスポンスの確認、書き換えができるネットワークキャプチャツール
- Telerik社がフリーソフトウェアとして公開
  https://www.telerik.com/fiddler

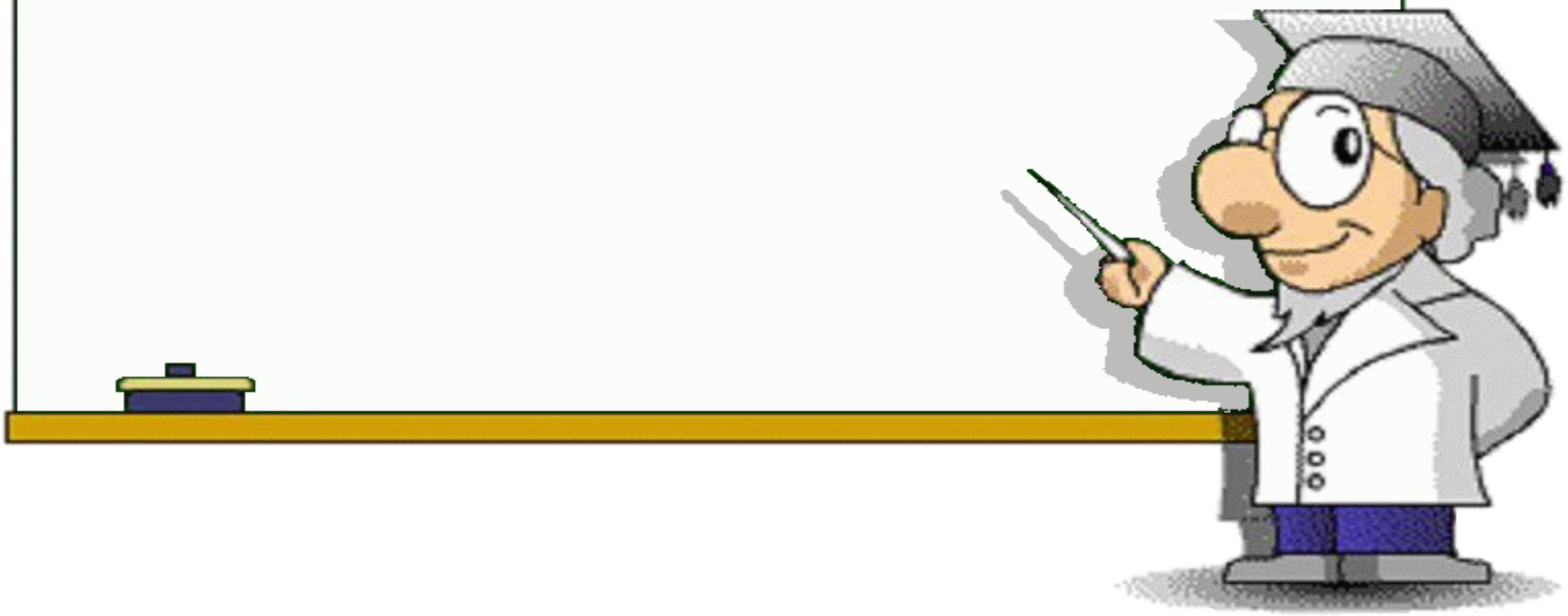
AppGoatには、HTTPのリクエストやレスポンスの確認や書き換えが必要な演習があり、その際にFiddlerを使用します。

本資料では、Fiddlerのインストール、設定、および基本的な操作について解説します。

※本資料の情報は2019年7月時点のものです。

内容

- Fiddlerのインストール
- 環境設定
- HTTPリクエスト/レスポンスの確認



Fiddlerのインストール

- Telerik社の「Download Fiddler」のウェブページにアクセスし、必要事項を記入してダウンロードします。
  https://www.telerik.com/download/fiddler

①Fiddlerを使用する目的を選択

②メールアドレスを入力

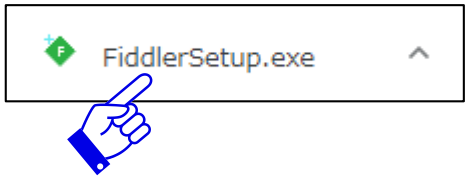
③国を選択

④チェックを入れる
（利用規約に同意）

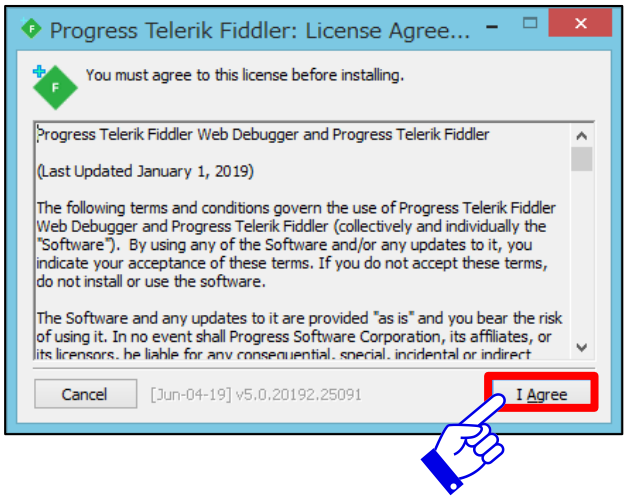
➡ Downloadボタンを押す

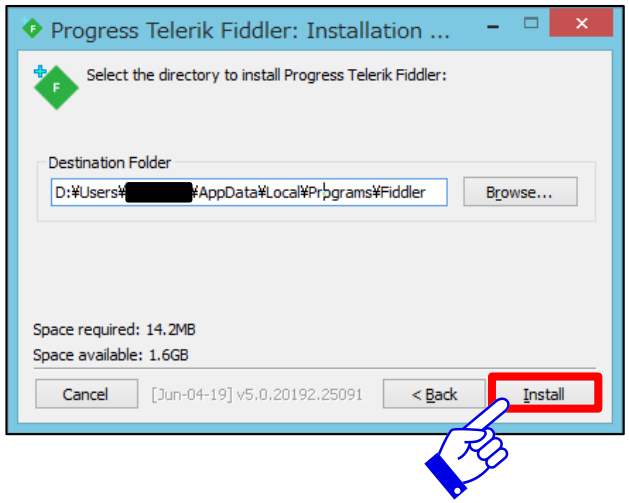
Fiddlerのインストール

- ダウンロードしたFiddlerSetup.exeをクリックして実行します。



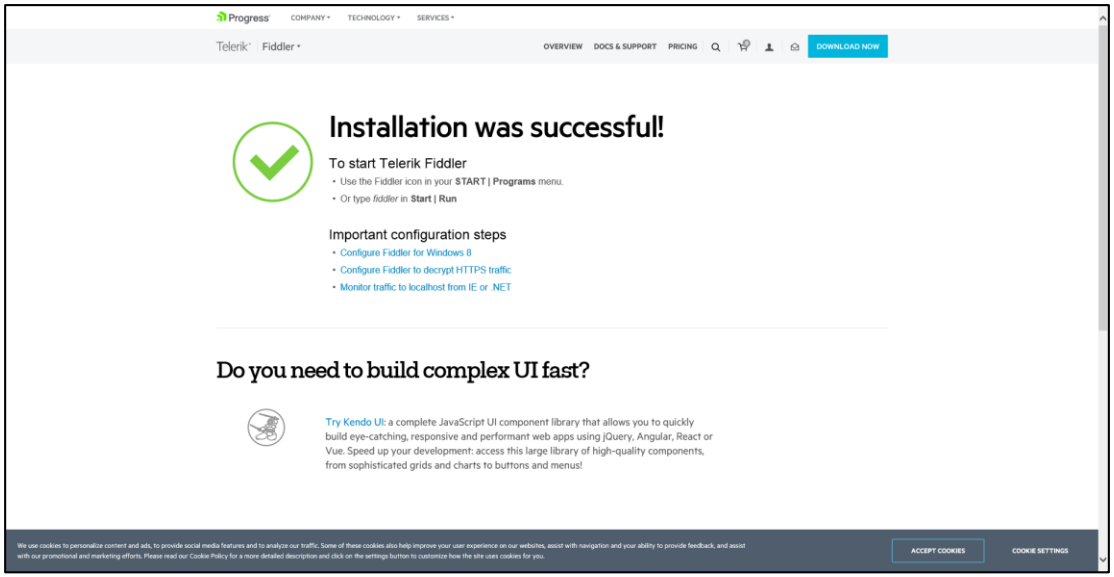
- インストールウィザードに従ってインストールしてください。





Fiddlerのインストール

- インストールに成功すると、「Fiddler 4」がメニューに追加されます。

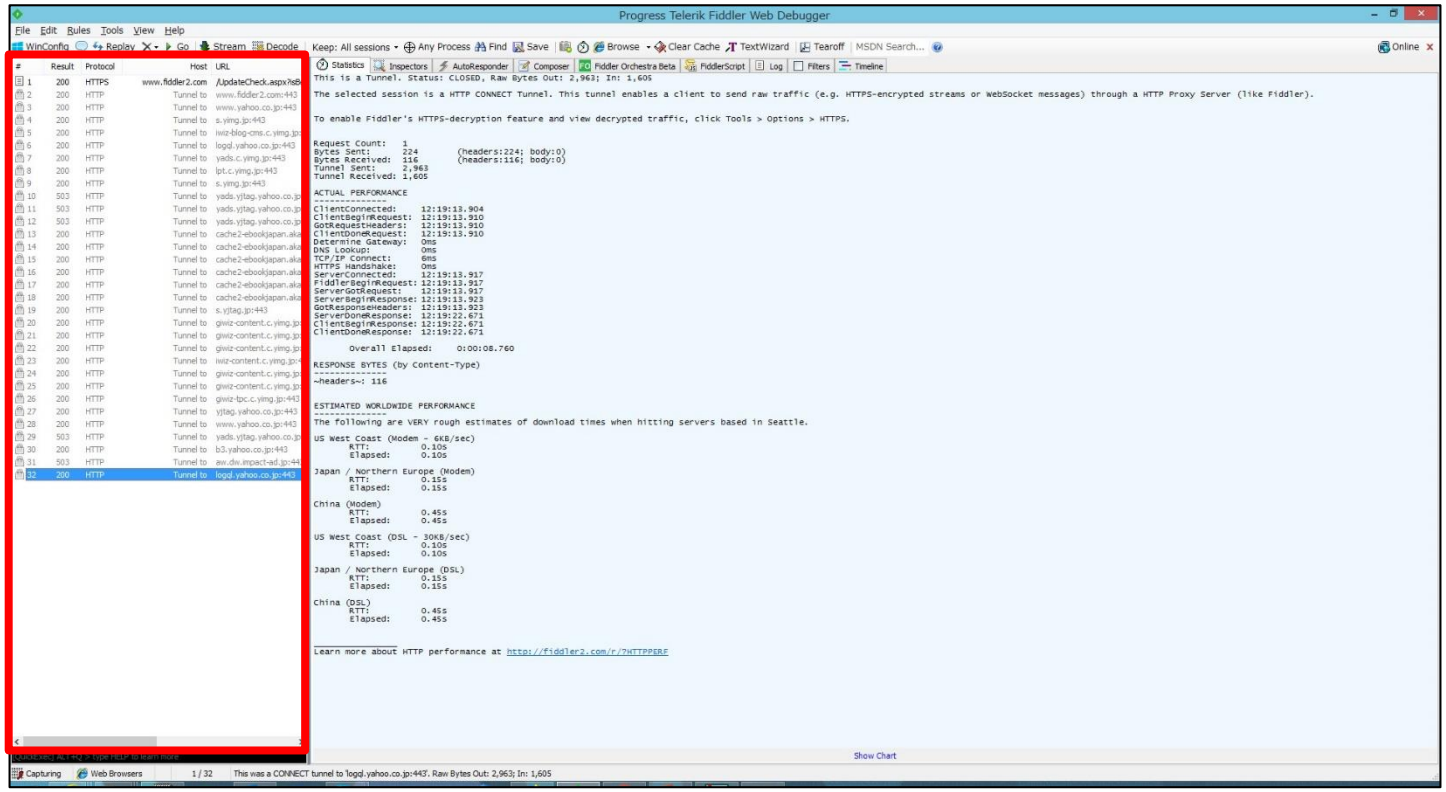


インストールが終了すると上の画面が表示されます。

環境設定

環境設定

- Fiddlerを起動し、その状態でブラウザで何らかのサイトにアクセスすると、HTTP通信がキャプチャされます。



環境設定

- 自動でキャプチャされない場合は、次の設定を確認してください。
  - 画面左下の「Capturing」を on にする。

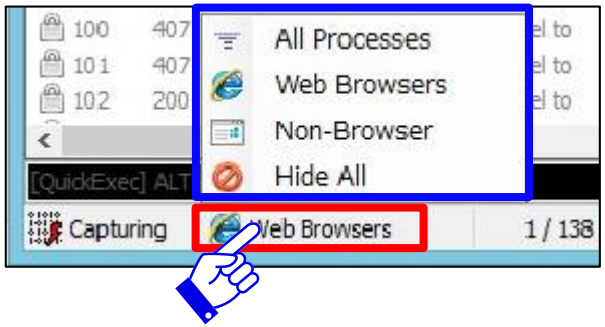
「Capturing」が off の状態

赤枠の部分をクリックする

「Capturing」が on の状態

環境設定

キャプチャ対象の設定



「Capturing」右側の
赤枠の部分をクリックすると、
キャプチャ対象を選択できます。

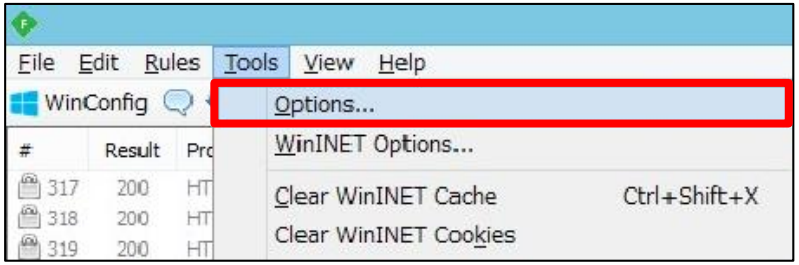
項目	説明
All Processes	全てのプロセスの通信をキャプチャ
Web Browsers	ブラウザの通信のみキャプチャ
Non-Browser	ブラウザ以外の通信をキャプチャ
Hide All	全てのキャプチャを隠す

ブラウザの通信をキャプチャするには「All Processes」または「Web Browsers」を選択します。

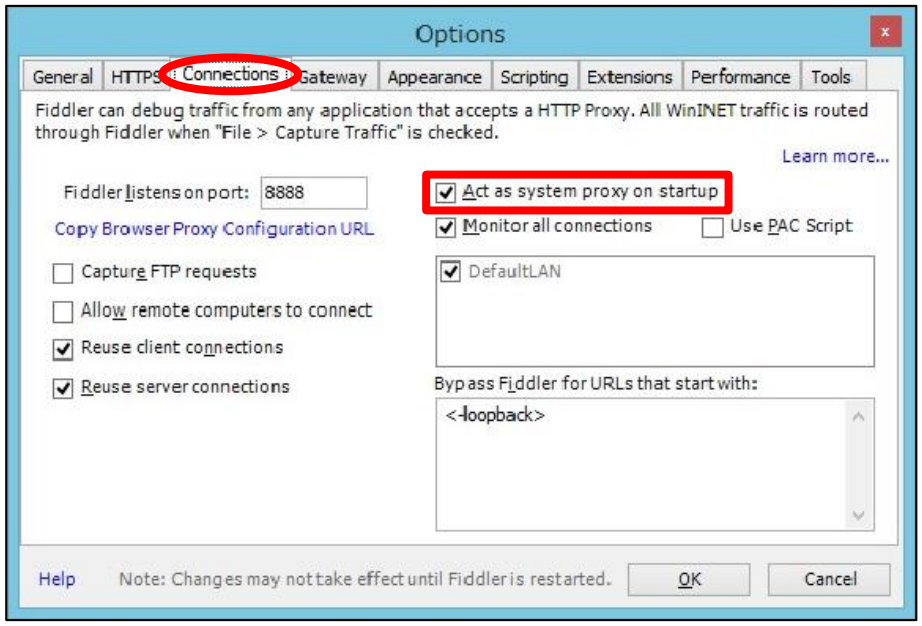
※AppGoatの演習で使用する場合には「Web Browsers」を推奨。

環境設定

Fiddlerのプロキシ設定



画面左上の「Tools」から「Options」を選択し、ダイアログを表示します。



「Connections」タブを選択し、「Act as system proxy on startup」にチェックが付いているか確認します。

チェックがない場合はチェックを入れて「OK」を押し、Fiddlerを再起動します。

環境設定

Firefoxのプロキシ設定

Firefoxの場合、インターネット接続に使用するプロキシ設定で「プロキシーを使用しない」を選択していると、Capturingをonにしていても通信がFiddlerを経由せずキャプチャができないため、設定を変更する必要があります。

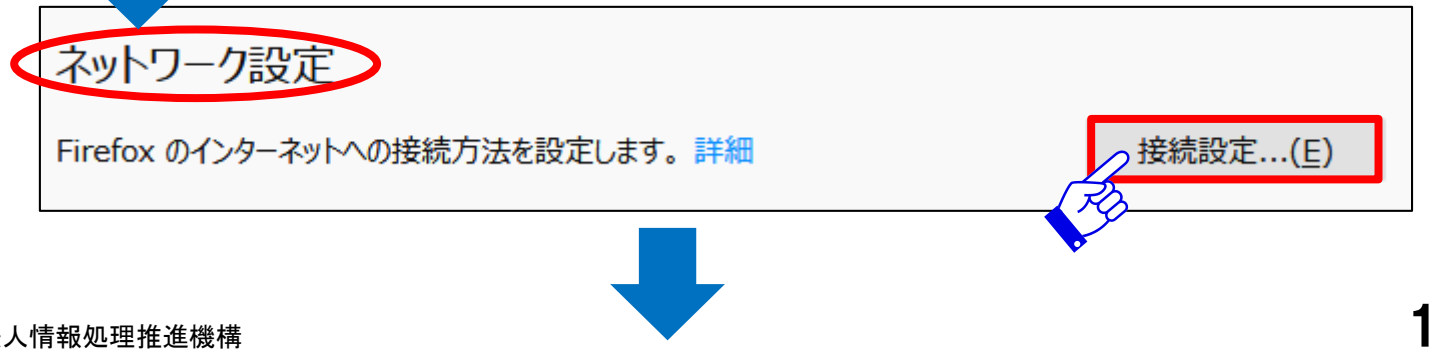
※設定を変更しても環境によってはキャプチャができない場合があることを確認しています。
その場合は他のブラウザの利用をご検討ください。

環境設定

Firefoxのプロキシ設定

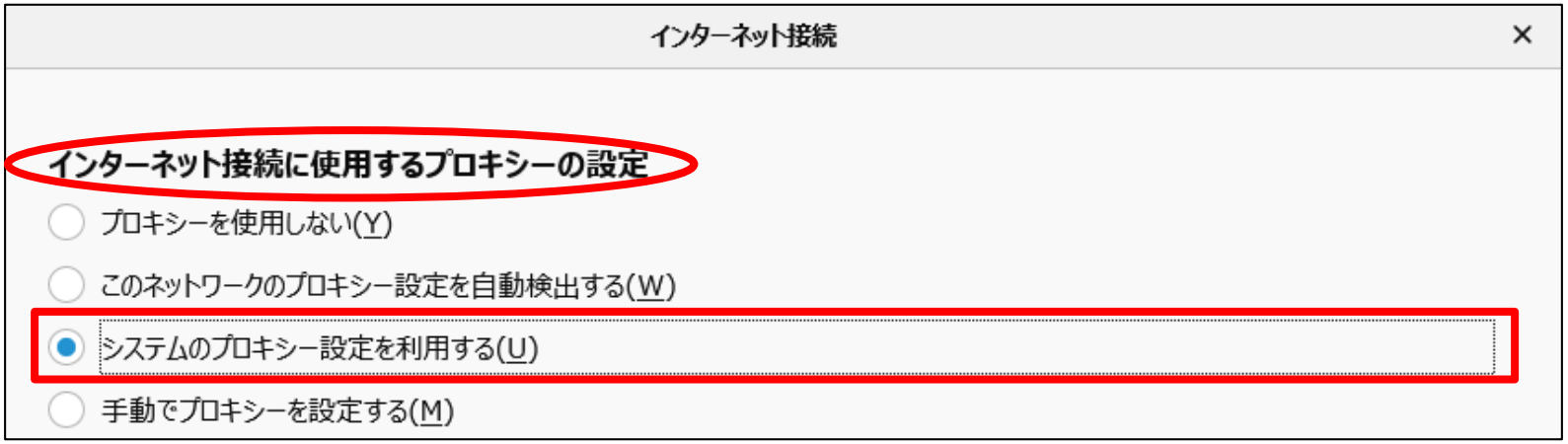
Firefoxの画面右上のメニューから「オプション」を開きます。

オプションページの一番下にある「ネットワーク設定」の「接続設定」を押すと、「インターネット接続に使用するプロキシーの設定」にて設定を確認・変更することができます。



環境設定

Firefoxのプロキシ設定



「インターネット接続に使用するプロキシーの設定」の
「システムのプロキシー設定を利用する」を選択します。

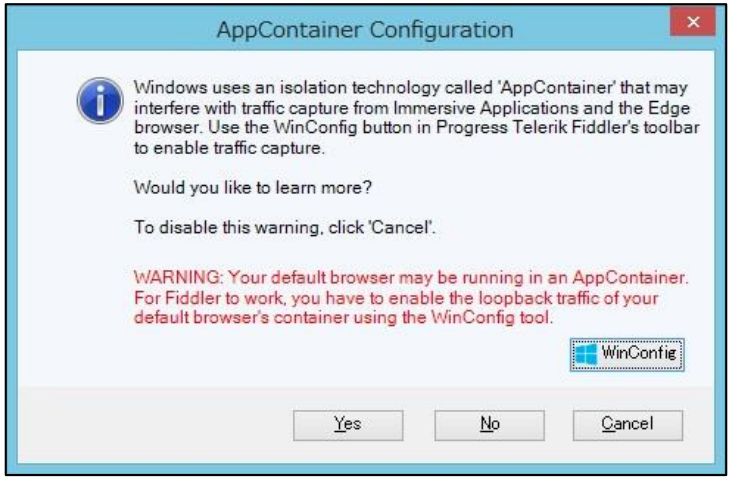
➡ 通信にFiddlerを経由するようになり、キャプチャが可能になります。

環境設定

- Windows 8.1 以降でキャプチャができない場合

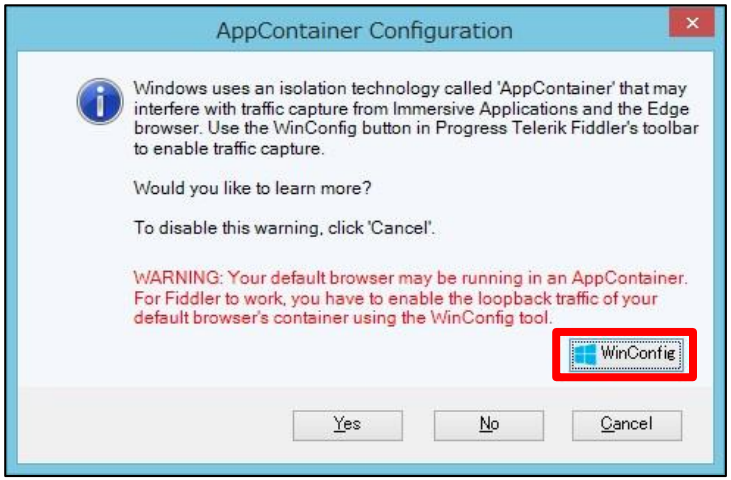
  Windows 8.1 以降には「AppContainer」という機能があり、Internet ExplorerやEdgeからFiddlerへの通信を妨げられてしまうため、設定をする必要があります。

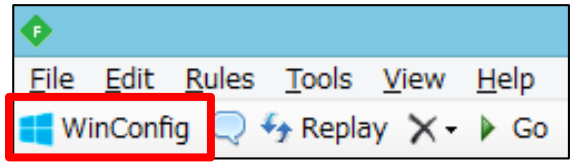
  該当するWindows環境の場合、Fiddlerの初回起動時にダイアログが表示されます。



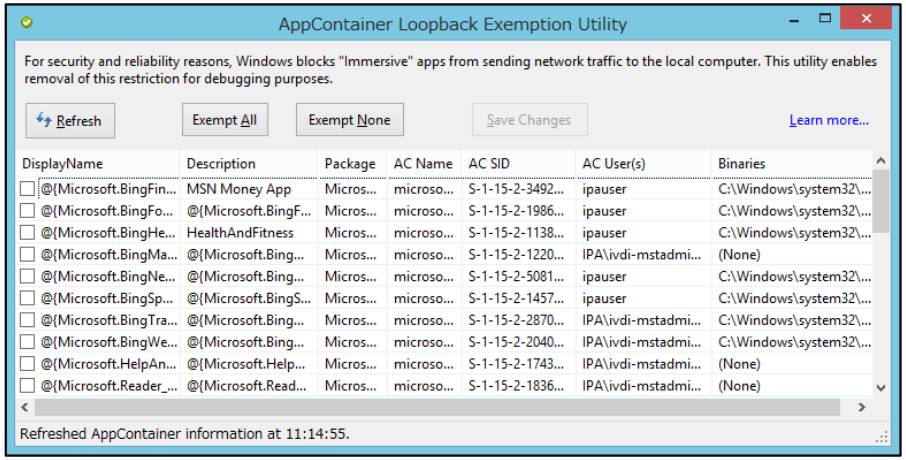
環境設定

Windows 8.1 以降でキャプチャができない場合



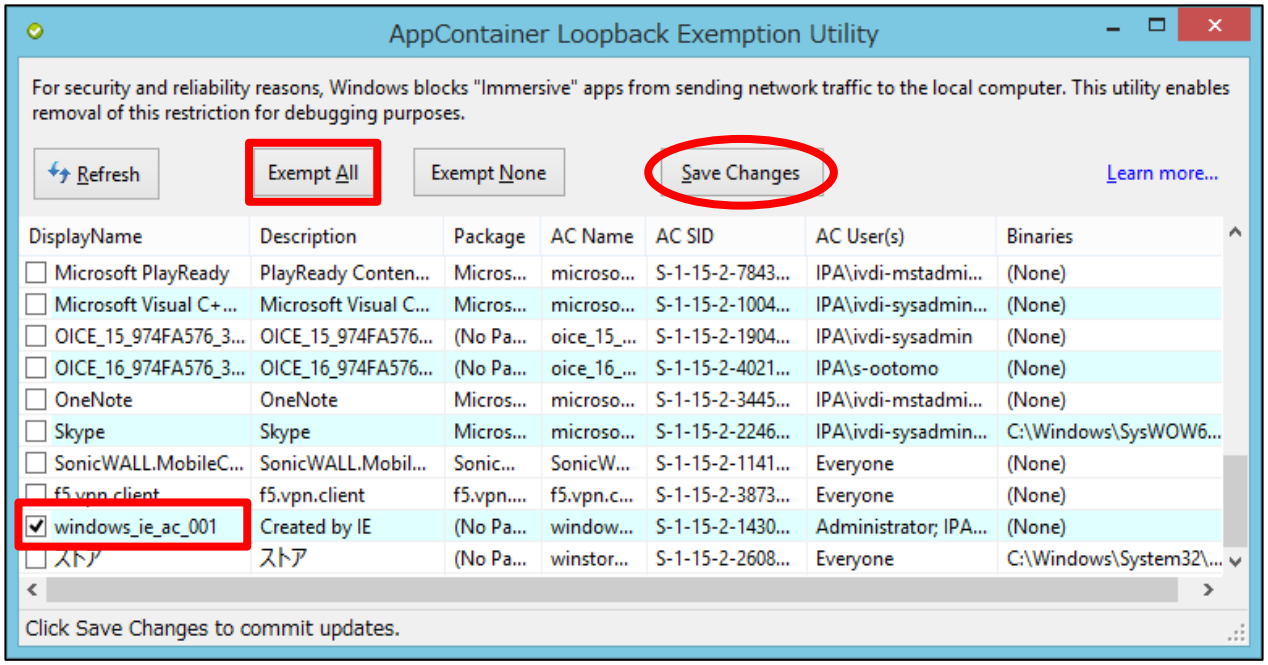


表示されたダイアログの右下、
またはFiddlerの画面左上の
「WinConfig」から、設定画面を
開くことができます。



環境設定

Windows 8.1 以降でキャプチャができない場合



「Exempt All」を押して全選択するか、「windows_ie_ac_###」にチェックを入れ、「Save Changes」を押して設定を反映させます。

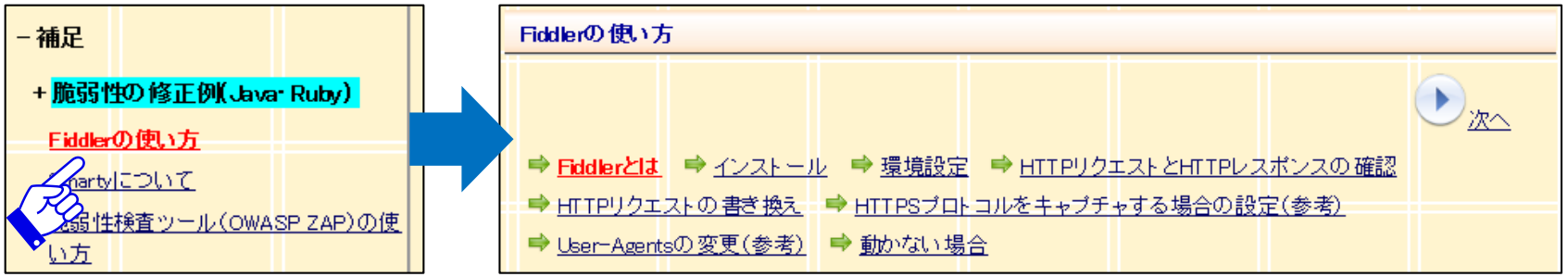
または前ページにおいて「WinConfig」とCtrlキーを同時に押すことで、全選択の状態が即時反映されます。

HTTPリクエスト/レスポンスの確認

HTTPリクエスト/レスポンスの確認

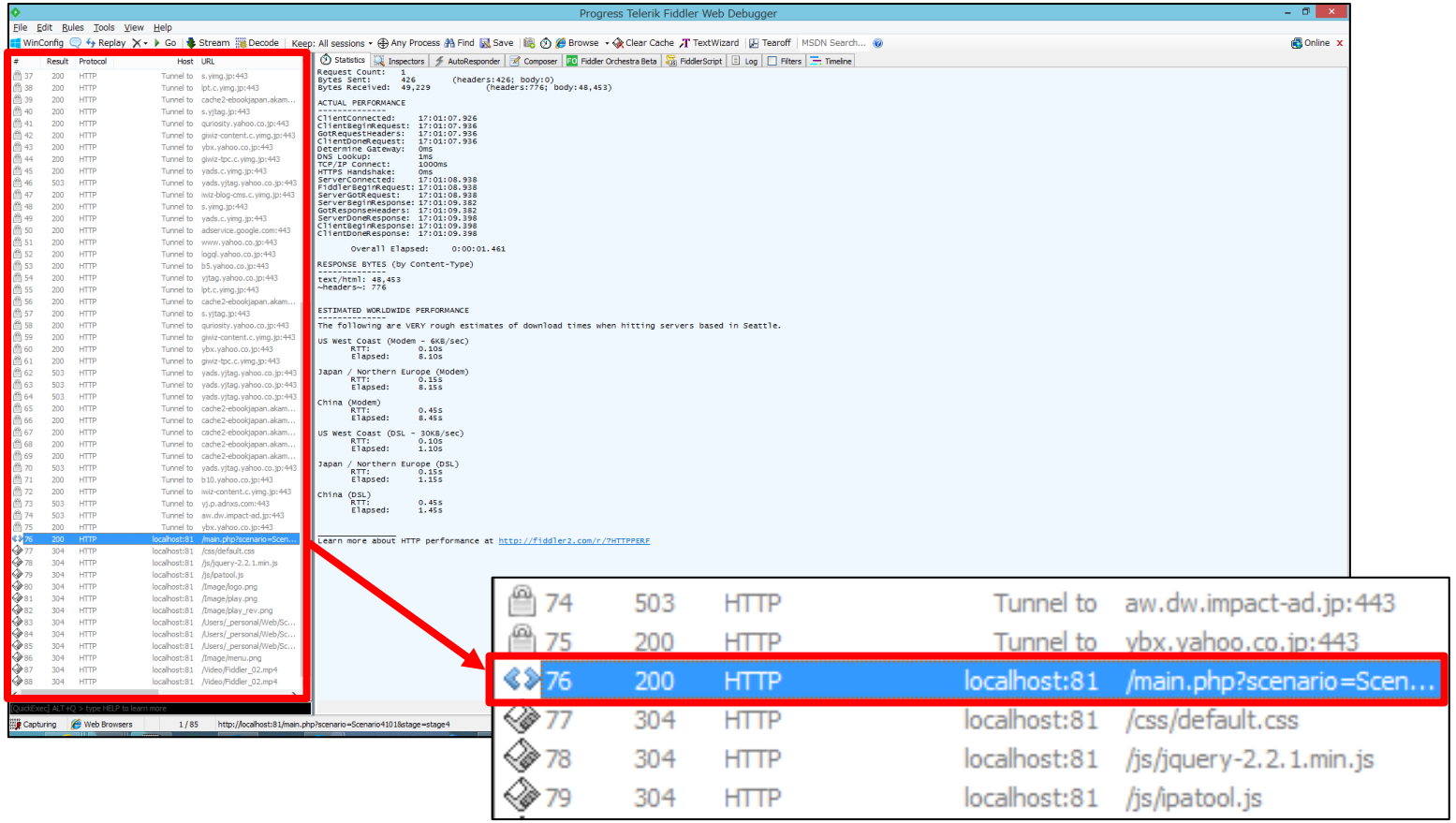
- 本資料では、AppGoatの「基礎」の演習で必要となる、HTTPリクエスト/レスポンスの確認方法を解説します。

- その他のFiddlerの機能である、HTTPリクエストの書き換えやHTTPS通信のキャプチャについては、AppGoatの「補足」の項目の中の「Fiddlerの使い方」を参照してください。



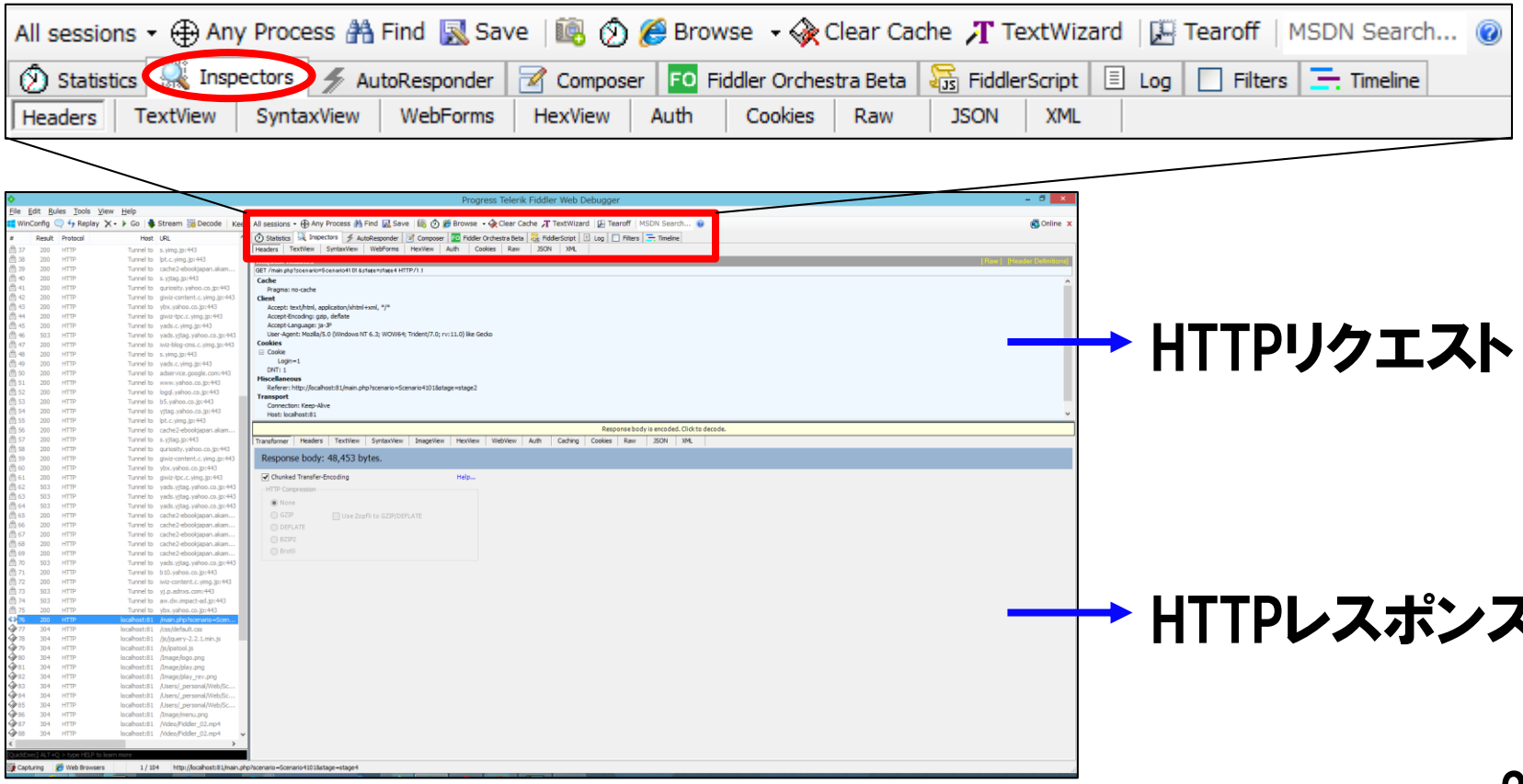
HTTPリクエスト/レスポンスの確認

- 左側のセッションリストから目的の通信を選び、クリックします。



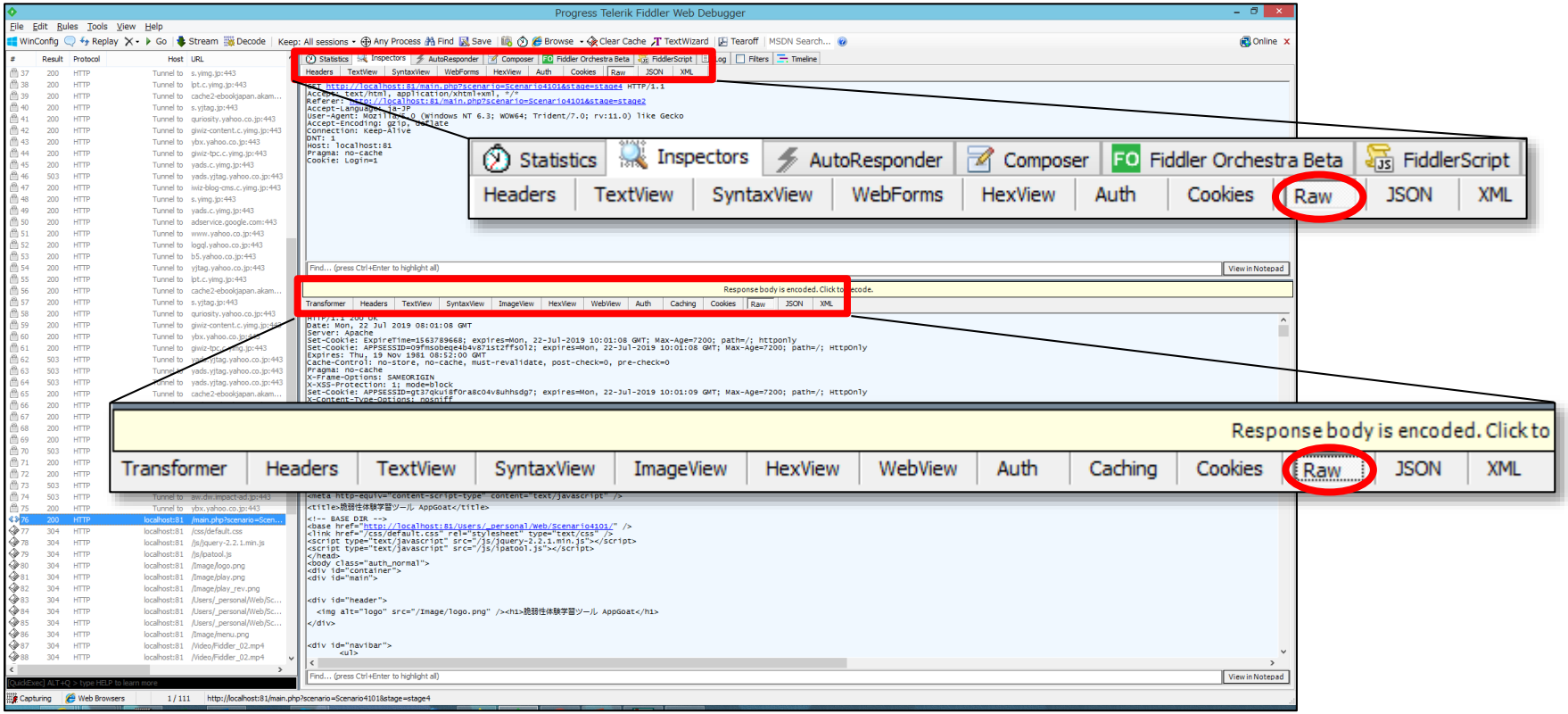
HTTPリクエスト/レスポンスの確認

- 右側ビューの上部にある「Inspectors」をクリックすると、上半分にHTTPリクエスト、下半分にHTTPレスポンスが表示されます。



HTTPリクエスト/レスポンスの確認

- リクエスト/レスポンスの上部にあるタブから表示形式を選択します。
- AppGoatの演習では「Raw」を使用します。



HTTPリクエスト/レスポンスの確認

- Cookie等の情報を確認することができます。

Transformer | Headers | TextView | SyntaxView | ImageView | HexView | WebView | Auth | C

```
HTTP/1.1 200 OK
Date: Mon, 22 Jul 2019 08:01:08 GMT
Server: Apache
Set-Cookie: ExpireTime=1563789668; expires=Mon, 22-Jul-2019 10:01:08 GMT; Max-
Set-Cookie: APPSESSID=09fmsobeqe4b4v871st2ffsol2; expires=Mon, 22-Jul-2019 10:
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Set-Cookie: APPSESSID=gt37qkui8f0ra8c04v8uhhsdg7; expires=Mon, 22-Jul-2019 10:
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

bd38
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.
<html lang="ja" xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<meta http-equiv="content-style-type" content="text/css" />
<meta http-equiv="content-script-type" content="text/javascript" />
<title>脆弱性体験学習ツール AppGoat</title>
```

HTTPレスポンスを「Raw」形式で表示した例

以上で、
Fiddlerの使い方の解説は
終了です。