|  |
| --- |
|  |
| **Inzi NonAcitveX Develop Guide** |
| **Server Side Develop Giude (HY\_InziAgent 사용 설명서)**  인지소프트 최하영 연구원 |

|  |
| --- |
|  |

# NonActiveX를 위한 HY\_InziAgent 개요

NonActiveX는 기존의 웹브라우저(Internet Explorer 환경)에서 사용하던 ActiveX(OCX)를 모두 제거 하고  
기존 웹 화면과 같은 기능을 수행하는 EXE(실행파일)를 브라우저에서 실행, 브라우저와 스캔 EXE간의 통신, I/O를 가능하게 하여 웹에서 호출 한 EXE 수행 결과 값을 브라우저로 리턴 받을 수 있습니다.  
중간 통신의 EXE와 브라우저간의 직접적인 통신이 어려워 로컬 웹 서버를 이용해   
통신을 가능하게끔 합니다.

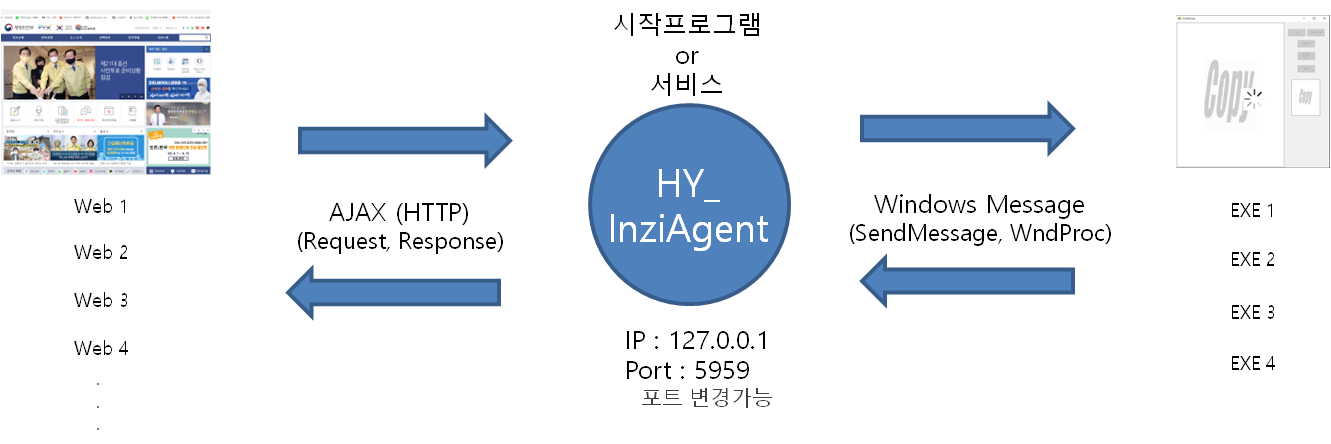
HY\_InziAgent는 WebSocket 방식이 아닌 HTTP 비동기(AJAX) 통신을 적용해야될 때 수행단에서 직접 Agent를 수정, 디버깅 할 수 있는 장점이 있습니다.

(행안부는 기존 ActiveX형태가 있던 것이 아니라 [As-Is : CS형식 – Delphi 개발]에서   
차세대 시스템이 웹기반으로 도입, 전환함에 따라 NonAcitveX 방식 도입)

브라우저 단에서는 EXE를 실행시키기 위해 custom URI를 사용하고 로컬 웹 서버와 Ajax 통신을 통하여 EXE와 통신 및 I/O작업을 할 수 있습니다.

제약사항 : OS – Windows 환경 Windows 7 SP2 (.NET 4.5)이상 사용가능  
프로젝트에서 사용하는 System.Threading.Tasks를 .net 4.5이상부터 지원  
브라우저 – 웹 표준을 준수하는 AJAX 지원 가능한 모든 브라우저 사용가능 (Internet Explorer 8 이상)

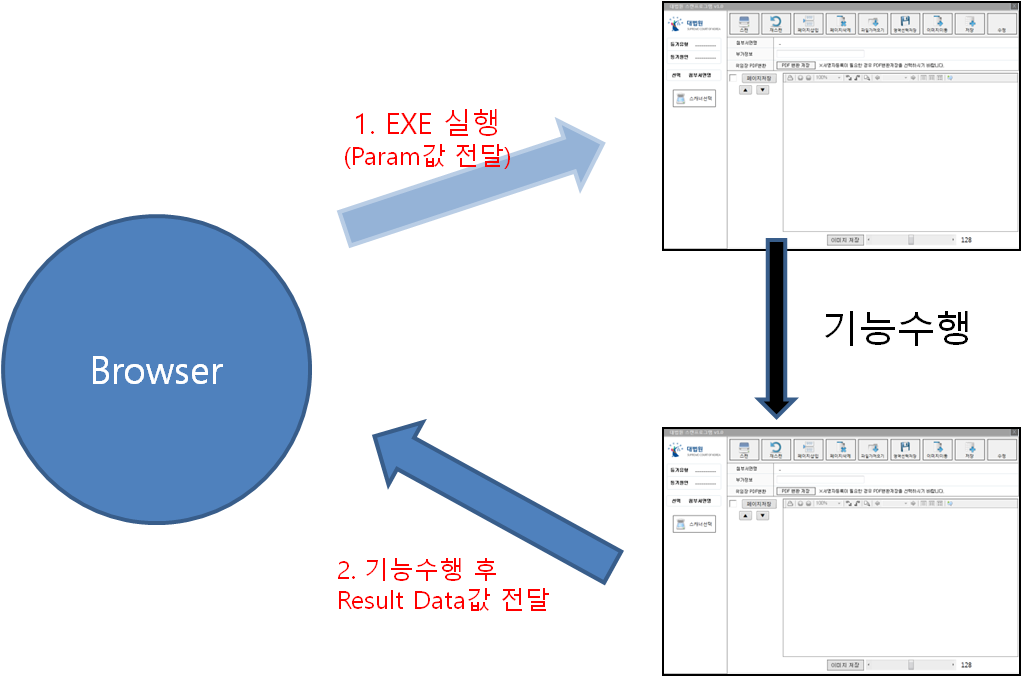
1. 구조



WEB과 EXE를 통신시키기 위해서는 사용자 PC에 Local Web Server(Inzi\_MoisAgent.exe)가 필요하며,  
이 Local Web Server(Inzi\_MoisAgent.exe)는 설치가 정상적으로 되었다면 윈도우 시작프로그램에 등록되어   
사용자가 컴퓨터를 실행할 때 자동부팅됩니다.  
(만약, 서버가 실행중에 오류가 생겨 종료되더라도, API를 통해 실행시킬 수 있습니다.)

1. 방법

브라우저단에서 inzi\_iFormScan NonActiveX를 사용하기 위한 단계는 다음과 같습니다.



1. EXE 실행 및 Param값 전달 **(Ajax – HTTP Request)**  
 EXE실행 및 Interface로 정의된 Param값을 전달   
Param값 : Json String2. EXE 작업 후 결과 값 콜백함수로 받아오기 **(Ajax – HTTP Response)** 콜백 인자 : 결과값(result) – json string

* 1번 단계에서 Agent를 사용 할 때에는 반드시 Interface객체 형식을 지켜야합니다.
* 1 – 2 반복 가능 (웹이랑 지속적 통신 가능)

1. 장점

- HTTP AJAX 통신방식을 사용하기 때문에 아주 오래된 브라우저에서도 작동 가능합니다.

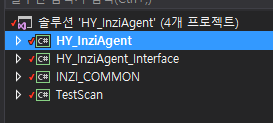
- 서버의 멀티스레딩 지원 - 웹에서 Request 요청에 대해 쓰레드 Task를 생성하므로 여러 개의 EXE에 대한비동기 처리가 가능합니다.

- WebSocket보다 리소스를 적게먹어 WEB과 간단한 통신을 할 때 사용하기 좋습니다.

- 서버 로그처리가 좋고 디버깅을 할 수 있습니다

- 커스터마이징하기에 용이합니다.

# 구성

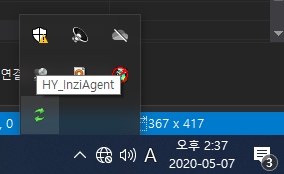


HY\_InziAgent : 로컬 웹 서버 소스  
HY\_InziAgent\_Interface : 로컬 웹 서버를 EXE에서 사용 할 때 쓰는 DLL (로컬 웹 서버 사용 DLL)  
INZI\_COMMON : Inzi iForm 공통  
TestScan : 테스트 스캔 프로그램 (EXE 개발시 참조)

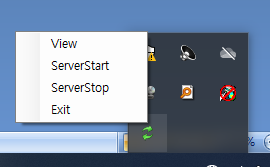
Agent에서 WEB에서 받고 보내는부분, EXE에서 받고 보내는 부분에 대한 소스는  
HY\_InziAgent 프로젝트의 **LOGIC 폴더**를 참조하세요

1. **HY\_InziAgent 화면 구성**

실행시 Notify Icon (Windows Tray)에 올라가게 됩니다.



우 클릭시 아래와 같이 목록이 나옵니다.



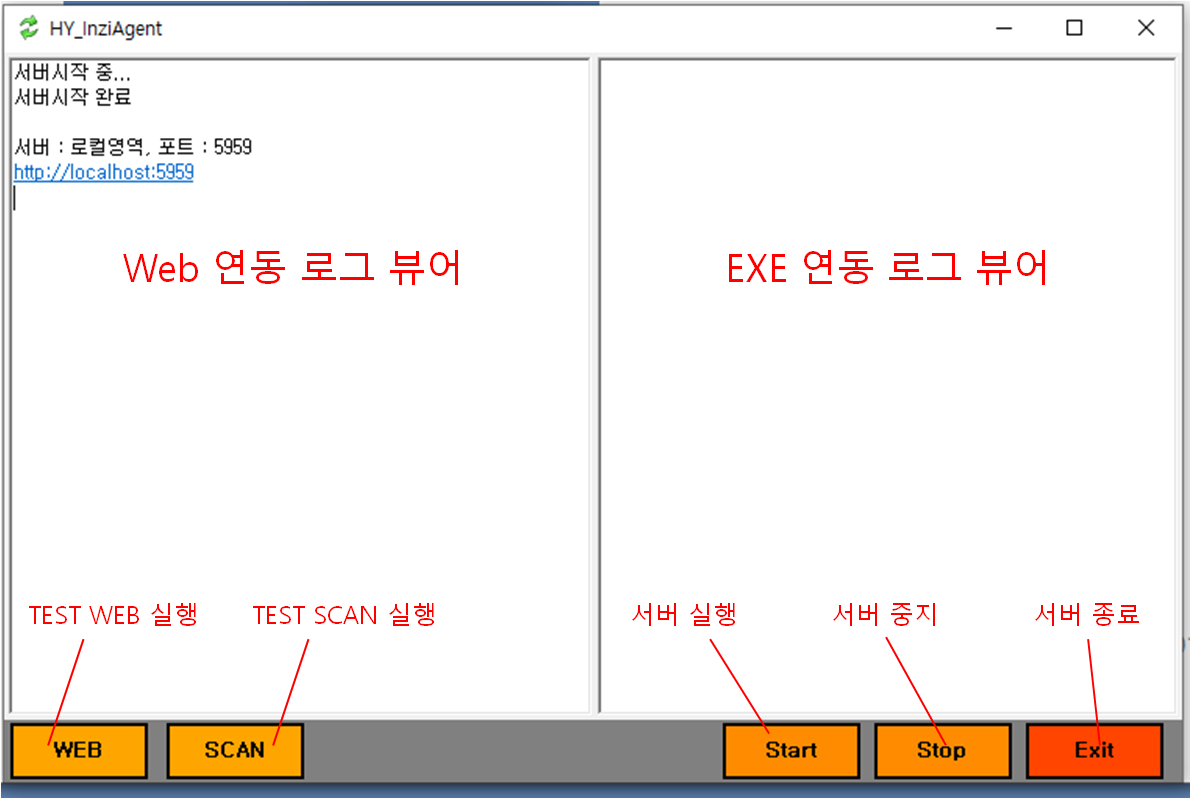
View : 로컬웹서버 화면보기

ServerStart : 서버 시작

ServerStop : 서버 중지

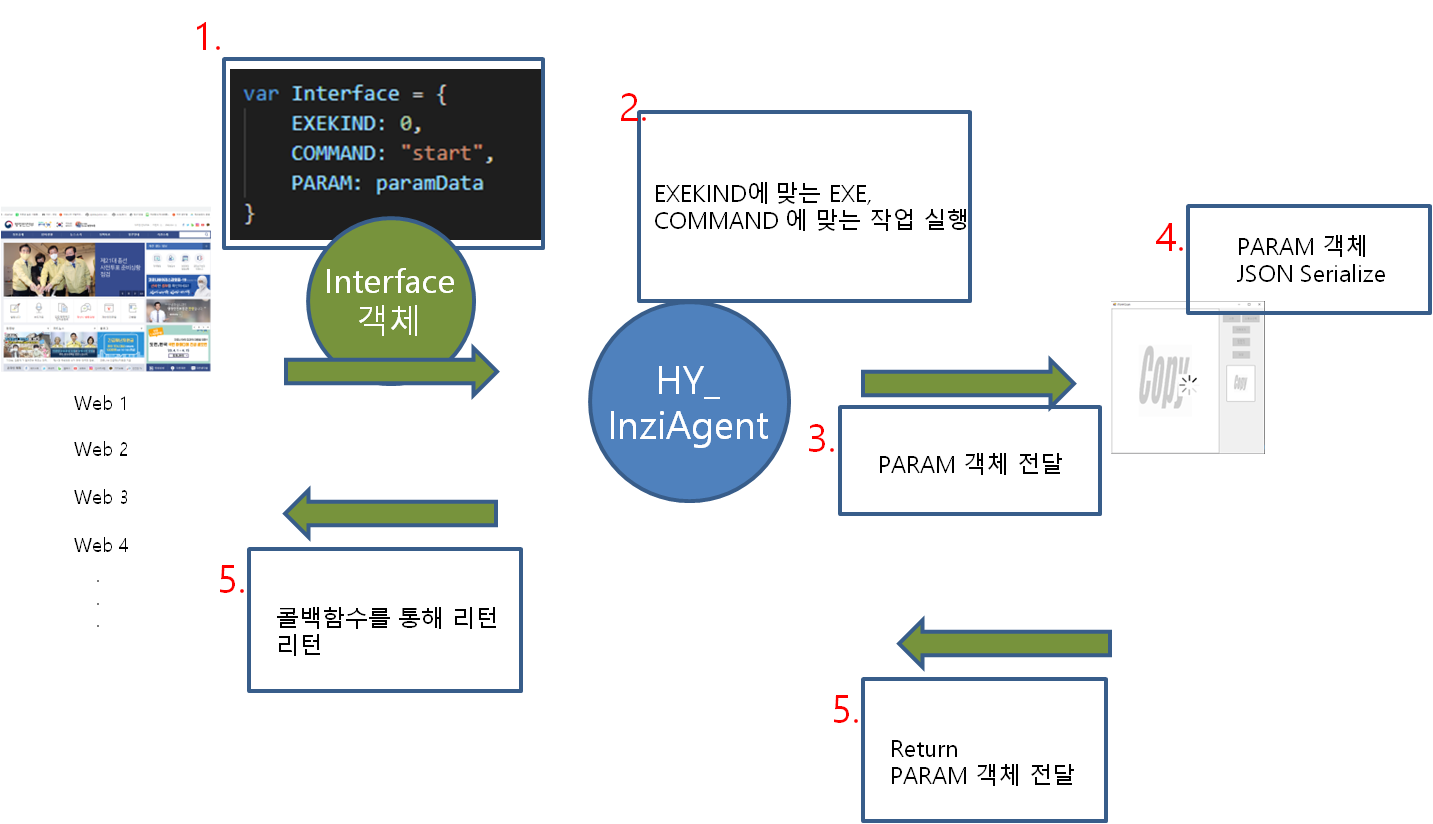
Exit : 로컬웹서버 종료

View 클릭시 다음과 같은 화면이 나옵니다.



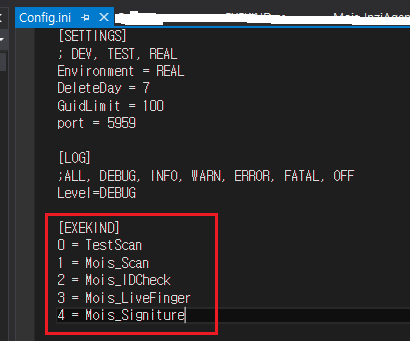
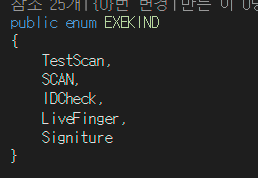
왼쪽 화면에서 Web과 Ajax 통신하는 로그를 확인.  
우측 화면에서 EXE와 Windows Message 통신하는 로그를 확인 할 수 있습니다.  
(우측상단 닫기버튼 누르면 NotifyIcon(Windows Tray)으로 되돌아갑니다)

1. **데이터 전달 흐름.**



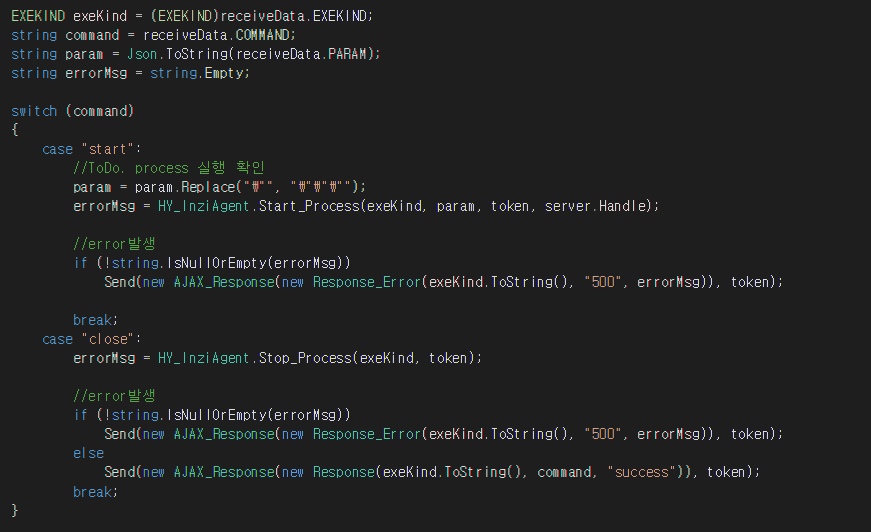
# 템플릿에 커스터마이징 개발이 필요한 부분

1. **EXE 종류, 구성 정하기**

* HY\_InziAgent INI 폴더 내부 Config.ini – EXEKIND  
  ****
* HY\_InziAgent\_Interface DATA 폴더 내부 EXEKIND.cs  
  

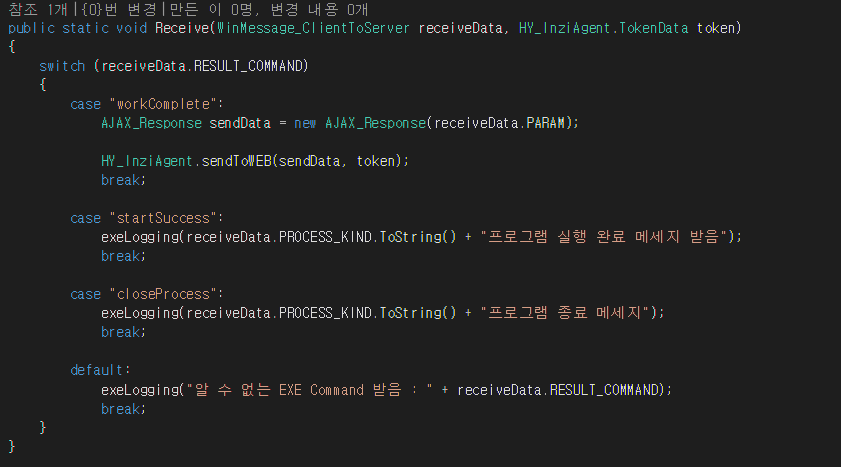
**EXEKIND enum의 순서와 Config.ini의 EXEKIND 순서를 맞춰야 함!**

1. **WEB으로 부터 받는 부분 (Request 처리부분)**Interface 객체 COMMAND 값 설계 필요

* Mois\_InziAgent LOGIC 폴더 내부 WEB.cs – Receive 함수  
  

COMMAND 설계 할 것

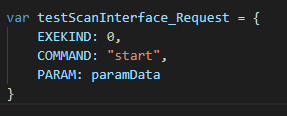
1. **EXE으로 부터 받는 부분**Return Interface 객체 COMMAND 값 설계 필요

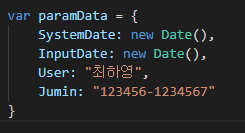
* Mois\_InziAgent LOGIC 폴더 내부 EXE.cs – Receive 함수  
  

RESULT\_COMMAND 설계 할 것

# Interface object

|  |  |  |
| --- | --- | --- |
| **Key** | **설명** | **Value 종류** |
| EXEKIND | 핸들링 할 EXE 종류를 선택합니다. [**Number** 값입니다.] | **0** : 테스트  **1**: 화상 스캔  **2**: 본인확인 **3**: 서명 **4**: 십지지문  **과 같이 실행 EXE 종류 설정** |
| COMMAND | EXE에 대한 명령을 내립니다.  [**String** 값입니다.] | **Start** : EXE를 실행시킵니다. **Close** : EXE를 종료시킵니다.  … 추가 생성 가능 HY\_InziAgent 프로젝트 LOGIC폴더 WEB 참조 |
| PARAM | 명령에 대한 인자값입니다. 반드시 각 EXE, COMMAND마다 정의가 되어 있어야 합니다. [**Object** 값입니다.]  EXE와 WEB간의 주고 받을 데이터를  정의해야 됩니다. | Ex)  {  param1 : value1,  param2 : value2,  param3 : value3 }  Or  Null |

Ex) Interface 객체 생성 예  


Param 값 생성 예  


# EXE개발 가이드 솔루션 내부에 포함된 TestScan 참조해주세요.

1. **실행시 EXEKIND값 저장시켜놓기**

전역변수로 EXEKIND enum 변수 저장.

1. **실행인자 받기 (1 – 서버핸들, 2 – Param)**
2. **인터페이스 처리 방법**
3. **서버로 신호 보내기**

# WEB 개발 가이드 - Mois\_InziJS.js (공통 js파일 API 설명)

**TESTWEB폴더에 있는 html파일을 참조하세요**

<script type="text/javascript" src="jquery-1.11.2.min.js" ></script>  
<script type="text/javascript" src="inziJS-1.0.js"></script>

1. 개요

• InziEXE\_Init() : Inzi\_MoisAgent 실행  
 • InziEXE\_GetReturn (interface, callbackFunc) : EXE 실행 (Interface 형식에 맞게) 및 callback함수 리턴  
 • SetEXETimeout (time) : InziEXE\_GetReturn() 요청 후 리턴값을 기다리는 최대 시간을 설정합니다.  
 • GetCustomURI() : Inzi\_MoisAgent를 실행시키기 위한 Custom URI값을 가져옵니다.

1. 상세

변수

**• INZI\_LocalServerURL**로컬웹서버 URL. Local이기 때문에 127.0.0.1 혹은 localhost

**• INZI\_LocalServerPORT**로컬웹서버 PORT번호 현재 5959

**• INZI\_LocalServer\_CustomURI**로컬웹서버를 실행시키는 Custom URI값. (사용하기 위해선 레지스트리에 등록되어 있어야 합니다.)

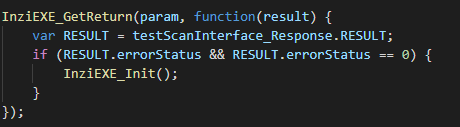
**• timeOutValue**Timeout 시간 값

메서드

**• InziEXE\_Init()**

로컬웹서버(Inzi\_MoisAgent.exe)를 실행시킵니다.  
**파라미터** – 없음

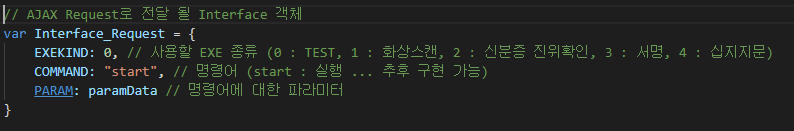
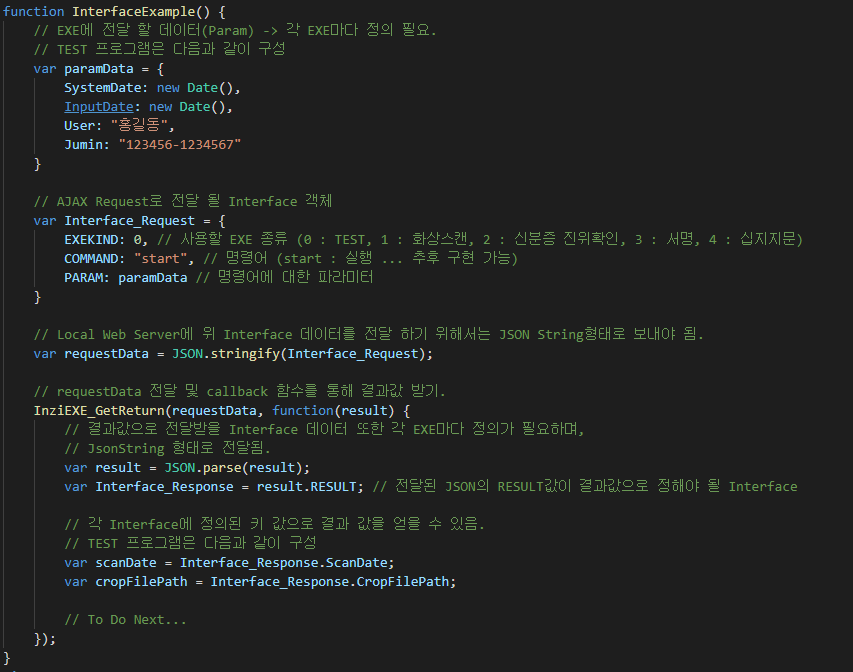
**리턴값** – 없음  
**사용방법** - InziEXE\_GetReturn()을 통해 Inzi 솔루션을 이용할 때 서버 응닶값이 없을경우 혹은   
{  
 errorStatus : 0,  
 errorMsg : “Server None”  
}  
과 같은 값이 떨어질 경우에 해당 API를 호출해서 Inzi\_MoisAgent.exe를 실행시켜줍니다

예시)  
 

**• InziEXE\_GetReturn(interface, callbackFunc)**

EXE를 Interface객체로 요청한 대로 작동합니다.(정해진 Interface형식에 맞게 전달해야 됨.)

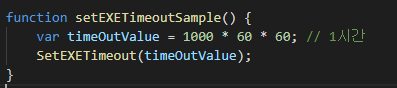
**파라미터** – interface : **정해진 Interface객체.** 자세한 사항은 [Interface항목](#_Interface_object) 참조.   
 callbackFunc : 요청 처리 한 후의 callback함수. – 인자값은 result 1개

**리턴값** – 없음  
**사용방법** - interface객체는 다음과 같은 형식을 유지해야됩니다.  
  
예시)

**• SetEXETimeOut(time)**

InziEXE\_GetReturn로 EXE 요청에 대한 리턴값을 기다리는 최대 시간을 설정합니다.  
EXE에 대한 리턴이 없을 경우 해당 정해진 시간이 지나면 InziEXE\_GetReturn의 콜백함수로 애러가 떨어집니다.  
Default 값은 30분입니다.

**파라미터** – time : 시간설정(ms)

**리턴값** – 없음  
사용방법 -   
예시)  


**• GetCustomURI ()**

Inzi\_MoisAgent.exe를 실행시키는 CustomURI값을 직접 얻습니다.

**파라미터** - 없음

**리턴값** – customURI string값

**• getBrowserPos()**

현재 브라우저의 좌표값을 가져오는 함수.

**파라미터** - 없음

**리턴값** - 좌표 string값 (“xpos,ypos,width,height”)

**• cSharpStringToJsString(cSharpString)**

C# String을 javascript String으로 바꾼다. (C#은 “ -> \” 와 같이 표현)

**파라미터** – cSharpString : 로컬웹서버로부터 전달받은 C# string값

**리턴값** - javascript string값

**• customURIExcute(customURI)**

CustomURI 실행

**파라미터** – customURI : customURI String값

**리턴값** – 없음