



# Magic Mirror API Documentation

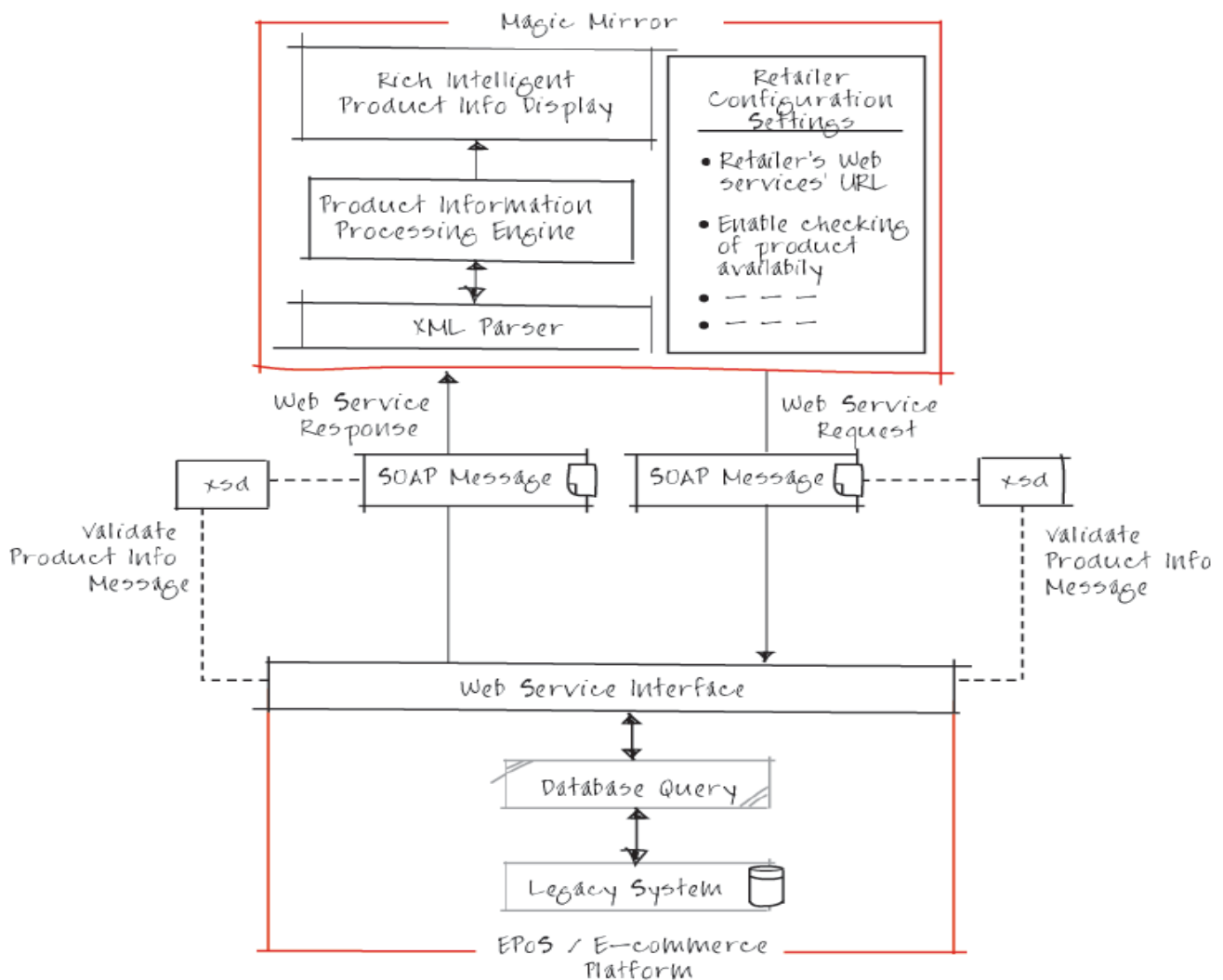
Version 1.0

# CONTENTS

Overview	2
1 Product Information Integration	3
1.1 Web Service Method Calls (Sample Implementation) .....	4
2 Stock Availability Integration	5
2.1 Web Service Method Calls (Sample Implementation) .....	6
3 EPOS or E-Commerce Integration	7
3.1 Web Service Method Calls (Sample Implementation) .....	8
4 Video Wall Integration	9
4.1 Web Service Method Calls (Sample Implementation) .....	10
5 Digital Signage System Integration	11
5.1 Web Service Method Calls (Sample Implementation) .....	12
6 Kiosk Integration	13
6.1 Web Service Method Calls (Sample Implementation) .....	14

# Overview

Magic Mirror is built on a truly flexible Service-Oriented Architecture (SOA) foundation. By implementing the Web Service that conforms to the published Web Services Description Language (WSDL), the Magic Mirror can be easily integrated to the retailer business environment to provide seamless customer experience and help to market their products. By applying this open framework, Magic Mirror can work with what the retailer already have in place and improve integration with lower cost and quicker deployments.

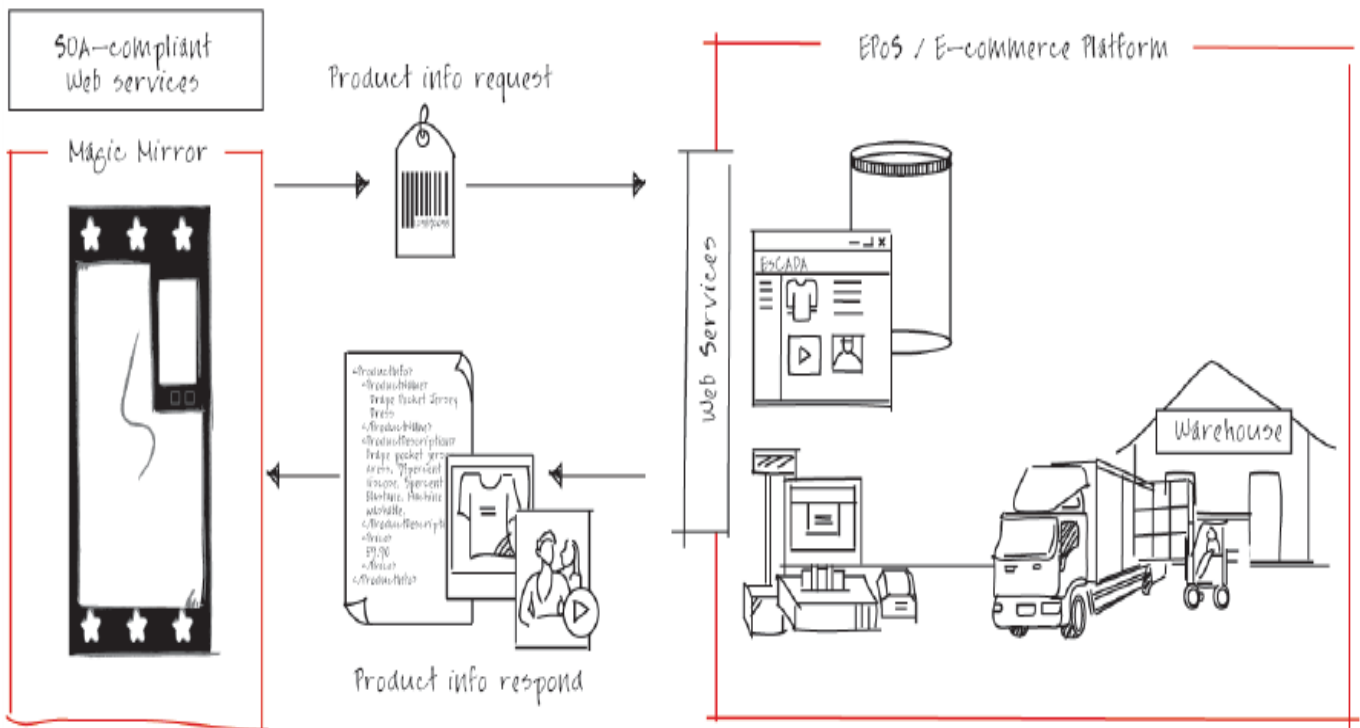


Technical Detail View of Magic Mirror and Retail's System Integration Framework

# 1 Product Information Integration

Magic Mirror can be linked to the retailer's ecommerce website to show a comprehensive list of products with detailed information. Recommendation engine can also be integrated in the mirror to suggest matching products with the selected item to create impulse buying.

When a product barcode is scanned using the scanner in Magic Mirror, a product info request is sent to invoke Retailer's Web Services. The product barcode is passed in as parameter and the required product info will be returned as a service call result. The result, which may include product name, product description, product images or videos, will be shown on the Magic Mirror intelligent rich media product info display.



*An Integration Overview of Magic Mirror and Retail's System*

## 1.1 Web Service Method Calls (Sample Implementation)

### **GetProductInfoByBarcode**

**Description:**

By passing in the product barcode, which is input through the Magic Mirror scanner, this method should return the information of the product via the returned parameters. All the parameters are nullable.

**Method Signature:**

```
public void GetProductInfoByBarcode(
    string barcode,
    out string productCode,
    out string productName,
    out string productPrice,
    out string productDescription,
    out string categoryCode,
    out string categoryName,
    out AttributeInfo[] productAttribute,
    out ProductInfo[] relatedItem,
    out ProductDisplayInfo[] productDisplay,
    out QuantityInfo[] productQuantity,
    out string buyItemURL,
    out string addToWishListURL)
```

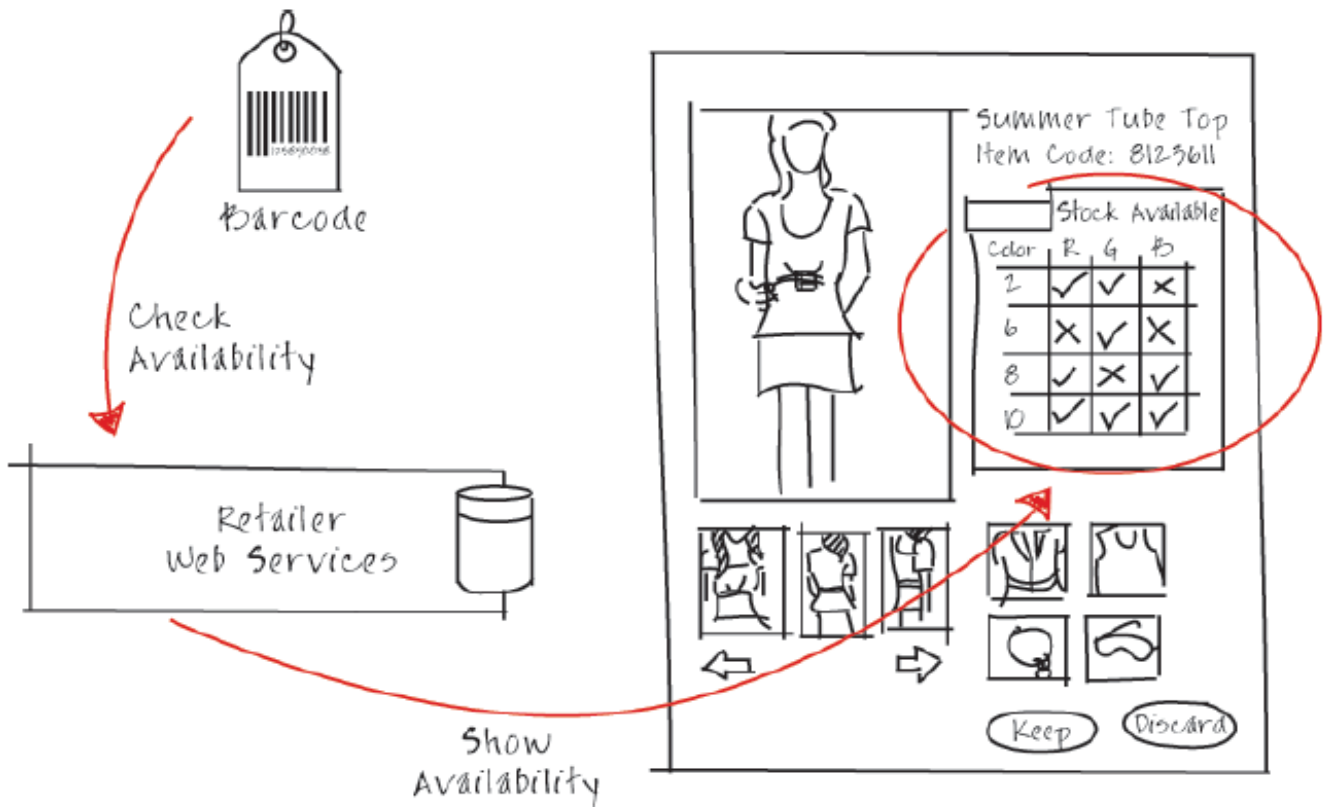
**Parameters:**

Parameter Name	Description
barcode	a pass-in parameter to search for product information.
productCode	return a nullable string for the product code.
productName	return a nullable string for the product name.
productPrice	return a nullable string for the product price.
productDescription	return a nullable string for the product description.
categoryCode	return a nullable string for the product category code.
categoryName	return a nullable string for the product category name.
productAttribute	return a nullable array of AttributeInfo complex type for the product attributes.

## 2 Stock Availability Integration

Retailers wouldn't require a physical place to store the whole collection of products with different colours or sizes to prevent sales lost due to stock unavailability in store. This would also help to increase the sales opportunity when customers are exposed to a higher variety of options.

Based on the product barcode input through the scanner, Magic Mirror can clearly show the user the stock status and availability of the product. By sending a check availability request along with the product barcode to the Retailer's Web Services, the size and colour matrix is then loaded, showing the user the exact level of stock across each dimension of that product.



*A Stock Availability Integration Overview*

## 2.1 Web Service Method Calls (Sample Implementation)

### **CheckAvailability**

**Description:**

This method is used to check for stock availability for products that are based on Size and Colour.

**Method Signature:**

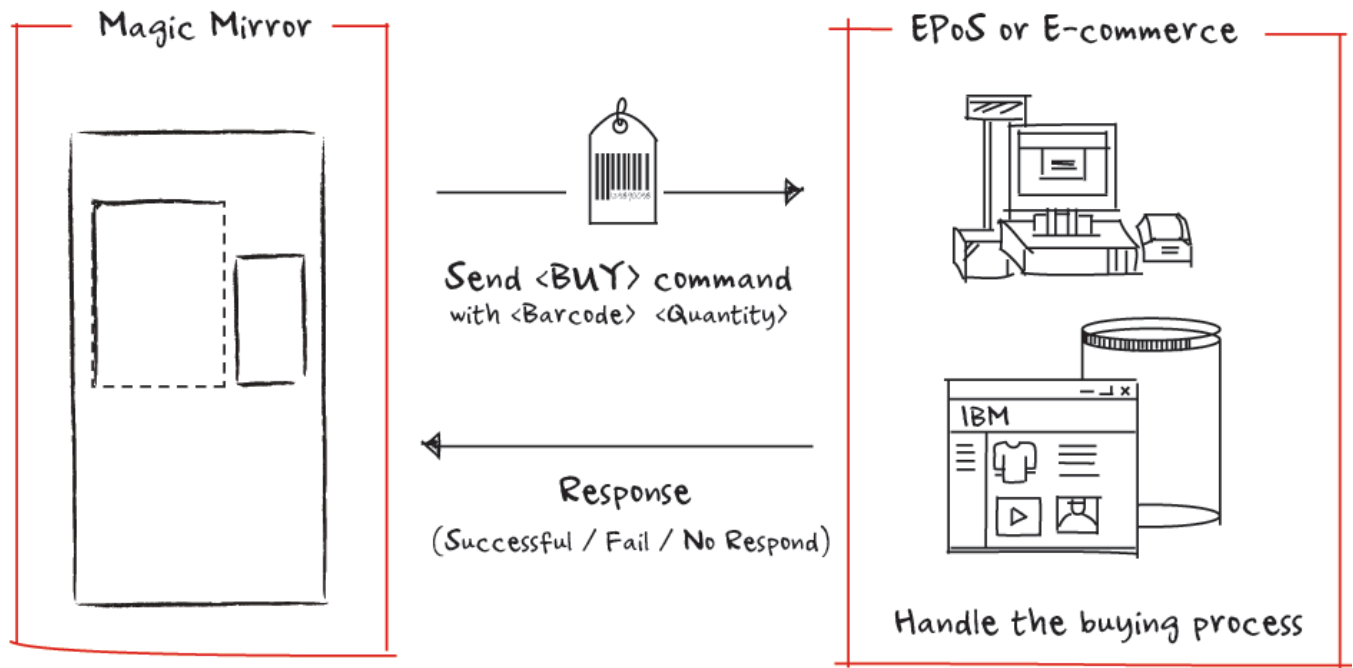
```
public void CheckAvailability(  
    string barcode,  
    string sizeID,  
    string colourID,  
    out double? quantity,  
    out Boolean? isAvailable)
```

**Parameters:**

Parameter Name	Description
barcode	a pass-in parameter for the product barcode.
sizeID	a pass-in parameter for the product SizeID.
colourID	a pass-in parameter for the product ColourID.
quantity	a nullable string returned parameter for the product quantity. A value of 0 or null indicates the product is out of stock.
isAvailable	a nullable string returned parameter for the product availability. A value of false or null indicates the product is out of stock.

### 3 EPOS or E-Commerce Integration

Not only display product information, Magic Mirror also can act as front end for EPOS or E-Commerce. Magic Mirror can send a “Buy” command along with Product Barcode and Quantity to the EPOS or E-Commerce System through an API or Web Service call. The EPOS or E-Commerce System will handle the buying process and return with response of successful, fail or no response.



*A Buying Process Integration Overview of EPOS or E-Commerce*



## 3.1 Web Service Method Calls (Sample Implementation)

### **BuyItem**

#### **Description:**

This method is used to send a buying command of a particular product to the E-Commerce or EPOS system.

#### **Method Signature:**

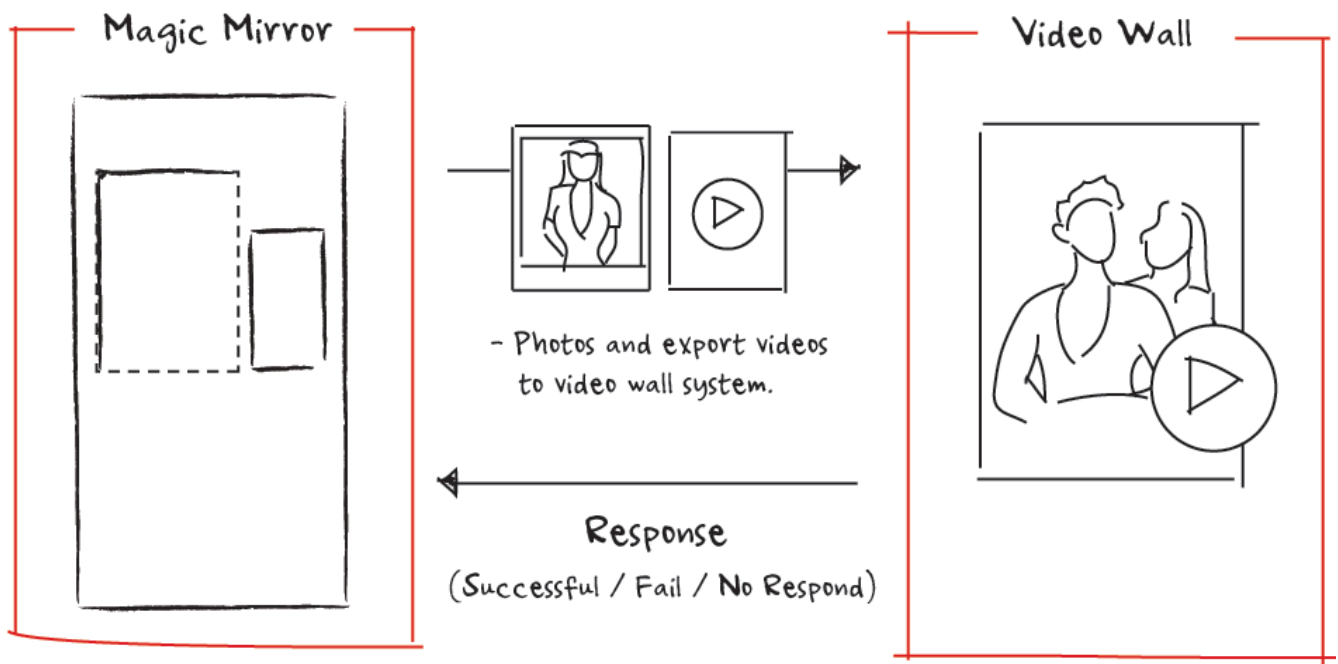
```
public void BuyItem(  
    string barcode,  
    double quantity)
```

#### **Parameters:**

Parameter Name	Description
barcode	a pass-in parameter for the product barcode.
quantity	a pass-in parameter for the product quantity. Default value should be 1.

## 4 Video Wall Integration

Magic Mirror can easily be integrated with any Video Wall Solutions. Photos and videos taken on Magic Mirror are scheduled to be exported to Video Wall System to entice customers from different locations with the latest promotion or ongoing event in store which could lead to sales maximization.



*An Integration Overview of Video Wall System*

## 4.1 Web Service Method Calls (Sample Implementation)

### **UploadFile**

#### **Description:**

This method is used to upload image or video to a Video Wall System.

#### **Method Signature:**

```
public void UploadFile(  
    byte[] file,  
    string fileName,  
    string fileType)
```

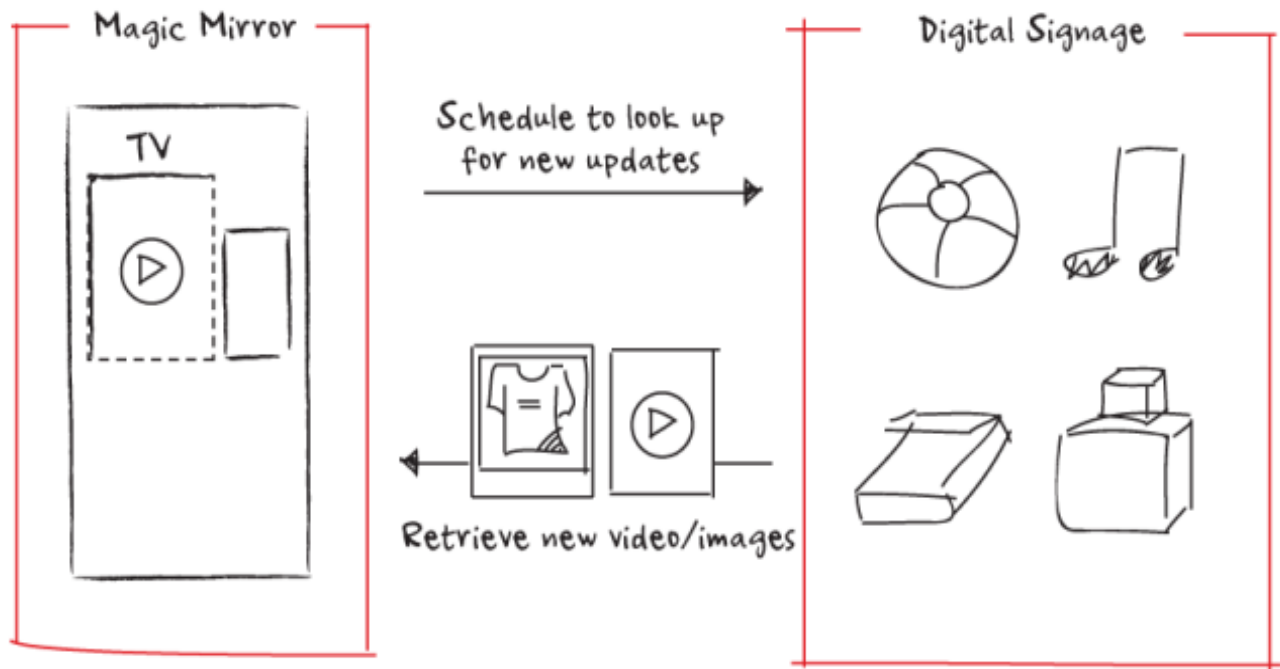
#### **Parameters:**

Parameter Name	Description
file	a pass-in parameter for the upload binary file.
fileName	a pass-in parameter for the filename.
fileType	a pass-in parameter for the file type, either Image or Video.

## 5 Digital Signage System Integration

Magic Mirror can be a digital signage which continuously plays marketing media to catch customers' eye towards the messages that retailers want to convey. It keeps the customers aware with the top selling items, latest season collection, latest contest, limited time promotion, etc. and encourages them to participate in the store events to maximize the returns.

Magic Mirror periodically looks up for the new media updates from Digital Signage System. If there are any updates on the Digital Signage System, the videos or images will be automatically synced to the Magic Mirror and displayed on the 40" LCD.



*An Integration Overview of Digital Signage System*

## 5.1 Web Service Method Calls (Sample Implementation)

### **MediaUpdate**

**Description:**

This method is used to retrieve image and video updates from the Digital Signage System.

**Method Signature:**

```
public List<Media> MediaUpdate ()
```

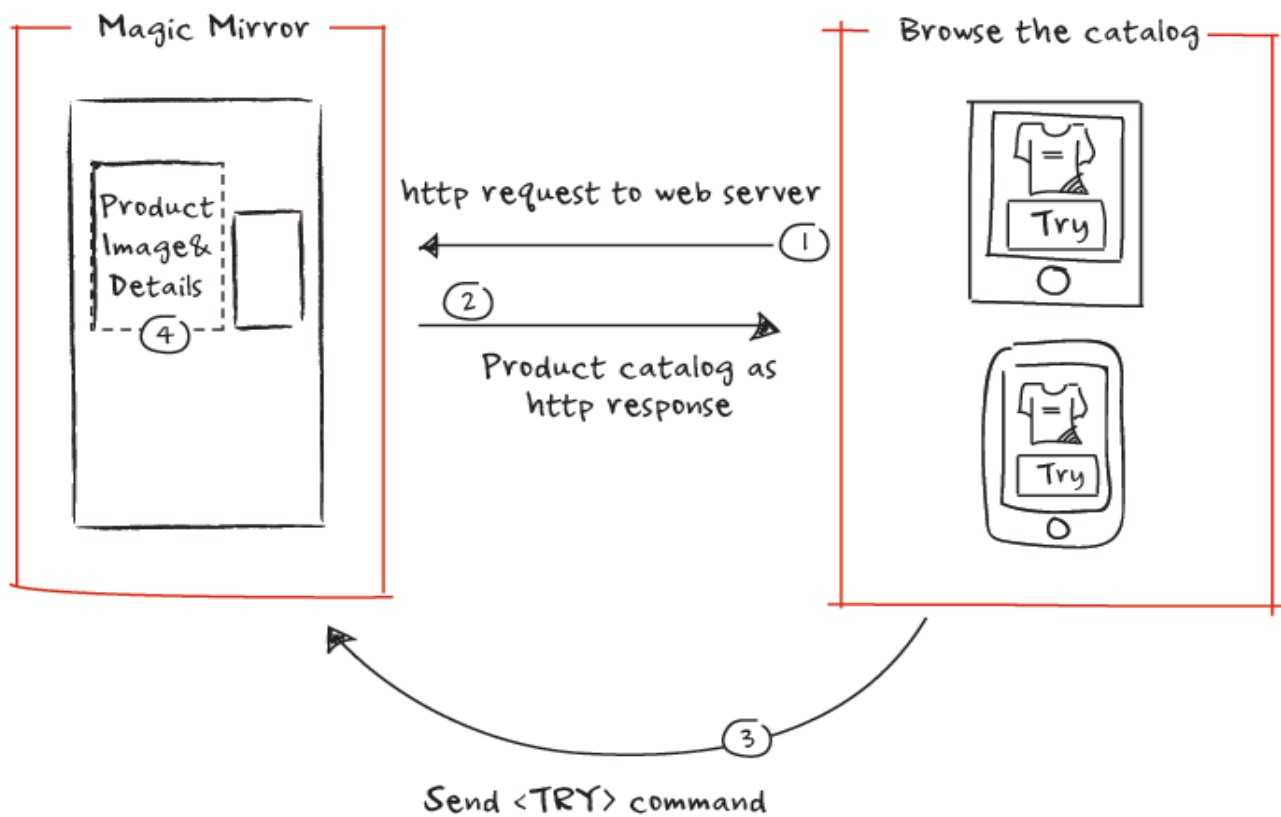
**ReturnValue:**

Return Type	Description
List<Media>	a return value of a list of images and videos

## 6 Kiosk Integration

Flipping through the items on iPad / iPhone is a more convenient and faster way in exploring the whole collection of the fashion store. By facilitating the shopping process, shoppers are more likely to shop and purchase in the retailer’s store.

User can browse the product catalog with iPad/iPhone by sending an HTTP request to the web server on Magic Mirror. By sending a “Try Now” command from iPad/iPhone to Magic Mirror, the item image and details will be displayed on the LCD Magic Mirror for the user to try on the item virtually. User could catch a glimpse on how the item is fitted on them and also minimize their time of trying on the unnecessary items. This may also create the desire of physically trying on the items because some of the dresses may not amaze shoppers when these dresses are hanging on the rack.



*An Integration Overview of Kiosk*

## 6.1 Web Service Method Calls (Sample Implementation)

### **BrowseProduct**

**Description:**

This method is used to retrieve a list of products to be displayed.

**Method Signature:**

```
public List<Product> BrowseProduct()
```

**Return Value:**

Return Type	Description
List<Product>	a return value of a list of product entities

### **TryItem**

**Description:**

This method is used to show a particular item details and image on the LCD screen.

**Method Signature:**

```
public void TryItem (  
    string barcode)
```

**Parameters:**

Parameter Name	Description
barcode	a pass-in parameter for the product barcode.