Building Apps and Extensions with MIT App Inventor

Evan W. Patton, Ph.D.

FOSDEM 2025 - 2025-02-02

MIT App Inventor by the Numbers

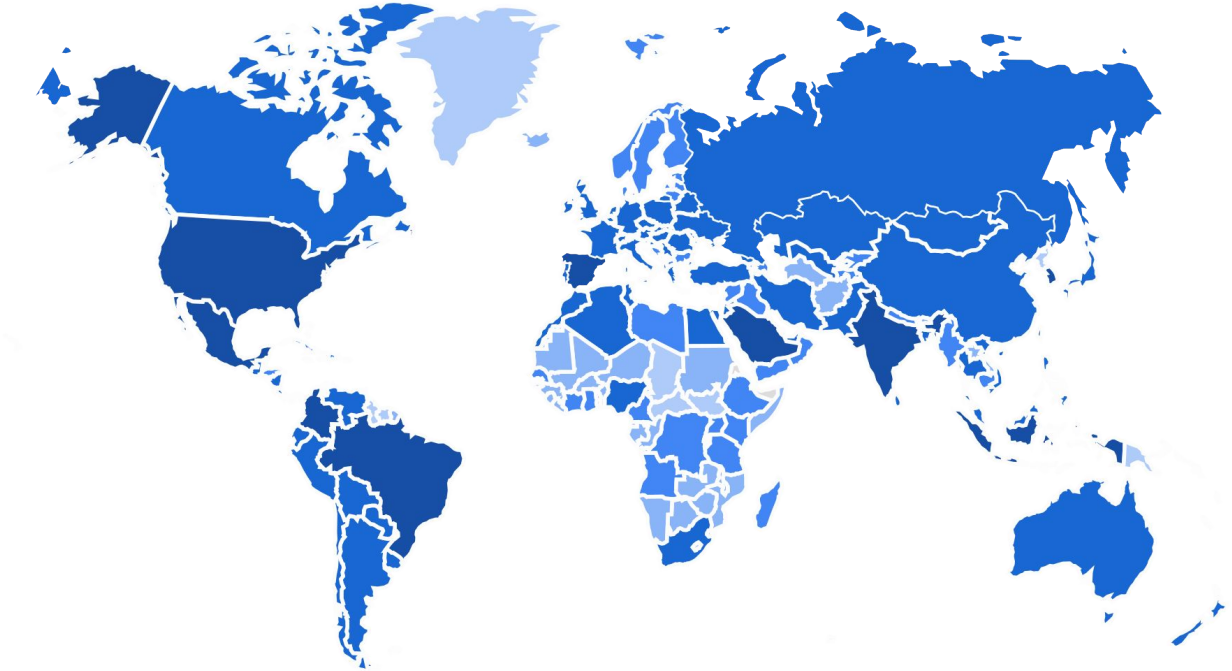
- Started in 2008 at Google, open sourced to MIT in 2011
- 23.4 million users since inception in 195 countries
- 113 million projects created
- 1.4 million users per month during peak usage

MIT
APP INVENTOR

The goal of computing education needs to move beyond computational thinking to a perspective of *computational action*. A computational action perspective on computing is founded on the idea that, while learning about computing, *young people should have the opportunity to do computing in ways that have direct impact on their lives and their communities.*

- Hal Abelson
Class of 1922 Professor of EECS
Director, MIT App Inventor

Computational Action Worldwide



United States	1291k
Taiwan	1125k
Mexico	883k
India	676k
Spain	625k

All time users: 23.4 million!

Computational Action

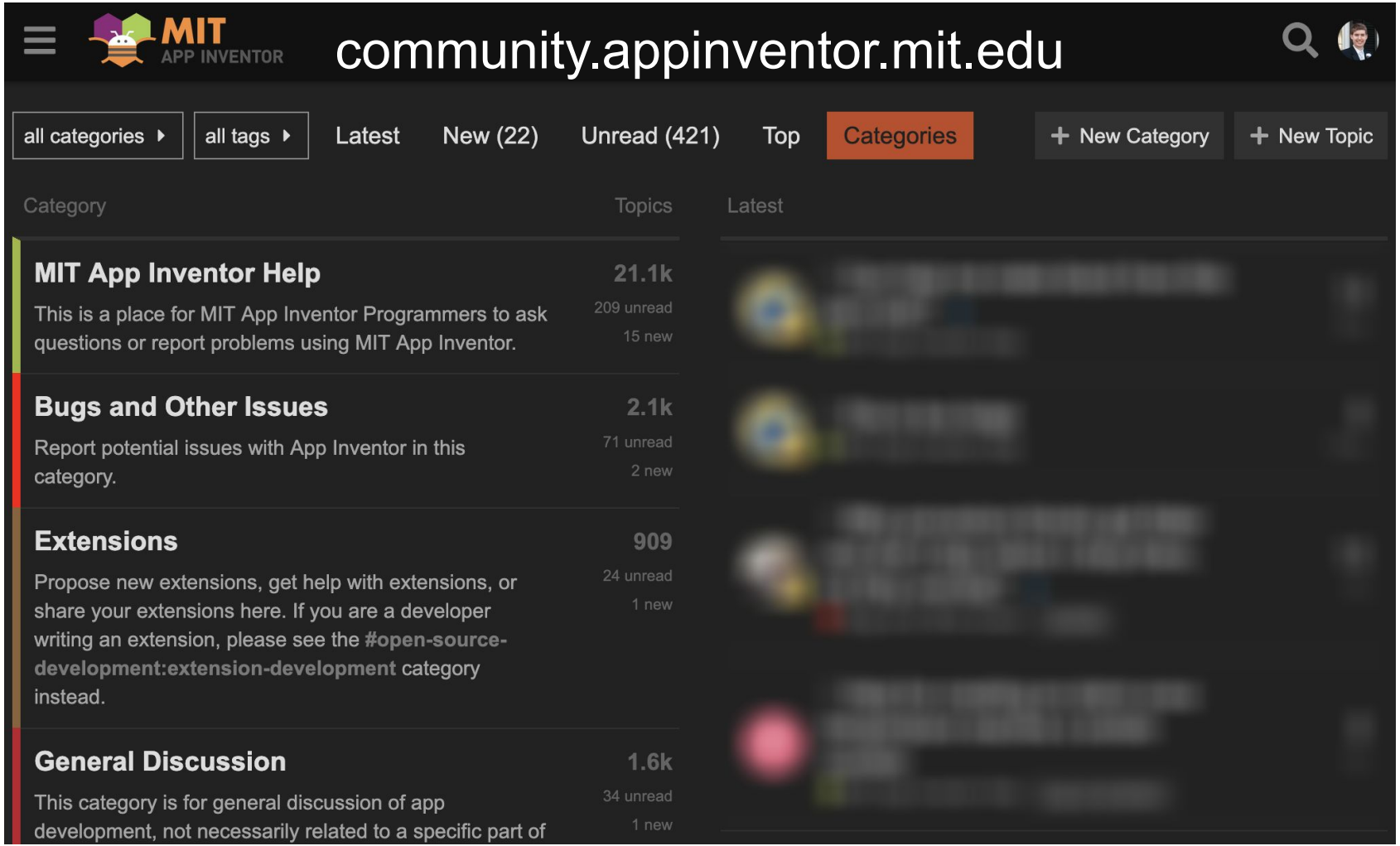
Global AI Hackathon: Adult Team Track Winners

App Name: *XôDengue*

Location: *Santa Catarina, Brazil*

Notable Technologies: *Google Teachable Machine Extension*

Mission: *Help the local community suffering from a dengue epidemic identify dengue-carrying mosquito larvae with 95% accuracy*


MIT
APP INVENTOR
community.appinventor.mit.edu
all categories
all tags
Latest
New (22)
Unread (421)
Top
Categories
+ New Category
+ New Topic
Category
Topics
Latest
MIT App Inventor Help
This is a place for MIT App Inventor Programmers to ask questions or report problems using MIT App Inventor.
21.1k
209 unread
15 new
Bugs and Other Issues
Report potential issues with App Inventor in this category.
2.1k
71 unread
2 new
Extensions
Propose new extensions, get help with extensions, or share your extensions here. If you are a developer writing an extension, please see the #open-source-development:extension-development category instead.
909
24 unread
1 new
General Discussion
This category is for general discussion of app development, not necessarily related to a specific part of
1.6k
34 unread
1 new

Demo

Extending App Inventor

- Written in Java
- Android only
- Usable in both live development and compiled apps
  - Extensions that add permissions are only usable in compiled apps

Extending App Inventor - Getting Started

Prerequisites:

- Java Development Kit 11 (e.g., openjdk)
- Apache Ant 1.10+

Options:

- Build in source tree
- Extension template
- Third party frameworks (e.g., Rush, Fast)

Extending App Inventor - Extension Anatomy

root directory

- aiwebres
  - icon.png - Icon shown in App Inventor
- assets
  - any resources needed by the code
- files
  - AndroidRuntime.jar - Class files used in final APK
  - component_build_infos.json - Metadata for final APK
- classes.jar - Android dex file for hot loading
- components.json - Block descriptions
- extension.properties - Extension metadata

Extending App Inventor - Building on Other Extensions

```
$ git clone -b extension/bluetoothle \
  https://github.com/mit-cml/appinventor-extensions
```

Extending App Inventor - Java Prologue

```
package edu.mit.appinventor.microbit;

import com.google.appinventor.components.annotations.*;
import com.google.appinventor.components.common.*;
import com.google.appinventor.components.runtime.*;
import com.google.appinventor.components.runtime.util.*;
import edu.mit.appinventor.ble.BluetoothLE;
import edu.mit.appinventor.ble.BluetoothLE.BluetoothConnectionListener;
import edu.mit.appinventor.ble.BluetoothLE.BLEResponseHandler;
import java.util.List;
```

Extending App Inventor - Define Extension Class

```
@DesignerComponent(version = 20250202,
    description = "My first BLE extension",
    category = ComponentCategory.EXTENSION,
    nonVisible = true,
    helpUrl = "https://example.com",
    iconName = "aiwebres/microbit.png")
@SimpleObject(external = true)
public class MicrobitExample extends AndroidNonvisibleComponent {
  public MicrobitExample(Form form) {
    super(form);
  }

  // Following slides
}
```

Extending App Inventor - Compiling

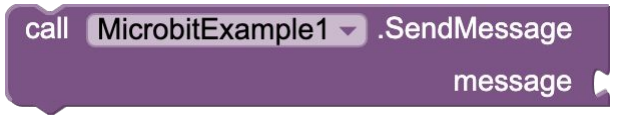
```
$ ant extensions
```

Extending App Inventor - Defining BLE Services

```
private BluetoothLE bleConnection = null;

private static final String SERVICE_UUID =
  "6E400001-B5A3-F393-E0A9-E50E24DCCA9E";
private static final String TX_UUID =
  "6E400002-B5A3-F393-E0A9-E50E24DCCA9E";
private static final String RX_UUID =
  "6E400003-B5A3-F393-E0A9-E50E24DCCA9E";
```

Component Blocks in App Inventor



Extending App Inventor - Defining an Event

```
@SimpleEvent
public void MessageReceived(final String message) {
  EventDispatcher.dispatchEvent(this, "MessageReceived", message);
}
```

Extending App Inventor - Defining an Event

```
private final BLEResponseHandler<String> readHandler =
  new BLEResponseHandler<>() {
    @Override
    public void onReceive(String service, String characteristic,
                          List<String> values) {
      String message = "";
      if (values != null && values.size() > 0) {
        message = values.get(0);
      }
      MessageReceived(message);
    }
  };
```

Extending App Inventor - Defining an Event

```
private final BluetoothLE.BluetoothConnectionListener listener =
  new BluetoothConnectionListener() {
    @Override
    public void onConnected(BluetoothLE bleConnection) {
      bleConnection.ExRegisterForStringValues(SERVICE_UUID,
          TX_UUID, false, readHandler);
    }

    @Override
    public void onDisconnected(BluetoothLE bleConnection) {
      bleConnection.ExUnregisterForValues(SERVICE_UUID,
          TX_UUID, readHandler);
    }
  };
```

Extending App Inventor - Defining a Property

```
@DesignerProperty(editorType =
    PropertyTypeConstants.PROPERTY_TYPE_COMPONENT +
        ":edu.mit.appinventor.ble.BluetoothLE")
@SimpleProperty
public void BluetoothDevice(BluetoothLE bluetoothLE) {
  if (bleConnection != null) {
    bleConnection.removeConnectionListener(listener);
  }
  bleConnection = bluetoothLE;
  if (bleConnection != null) {
    bleConnection.addConnectionListener(listener);
  }
}
```

Extending App Inventor - Defining a Property

```
/**
 * The BluetoothLE component connected to the micro:bit device.
 */
@SimpleProperty(category = PropertyCategory.BEHAVIOR)
public BluetoothLE BluetoothDevice() {
  return bleConnection;
}
```

Extending App Inventor - Defining a Function

```
@SimpleFunction
public void SendMessage(String message) {
  if (bleConnection != null) {
    bleConnection.ExWriteStringValuesWithResponse(SERVICE_UUID,
        RX_UUID, false, message, writeHandler);
  } else {
    reportNullConnection("SendMessage");
  }
}
```

Extending App Inventor - Error Handling

```
@SuppressWarnings("SameParameterValue")
private void reportNullConnection(String functionName) {
  form.dispatchErrorOccurredEvent(this, functionName,
      ErrorMessages.ERROR_EXTENSION_ERROR, 1,
      this.getClass().getSimpleName(), "BluetoothDevice is not set");
}
```

Extending App Inventor

```
@SimpleEvent
public void MessageSent(final String message) {
  EventDispatcher.dispatchEvent(this, "MessageSent", message);
}
```

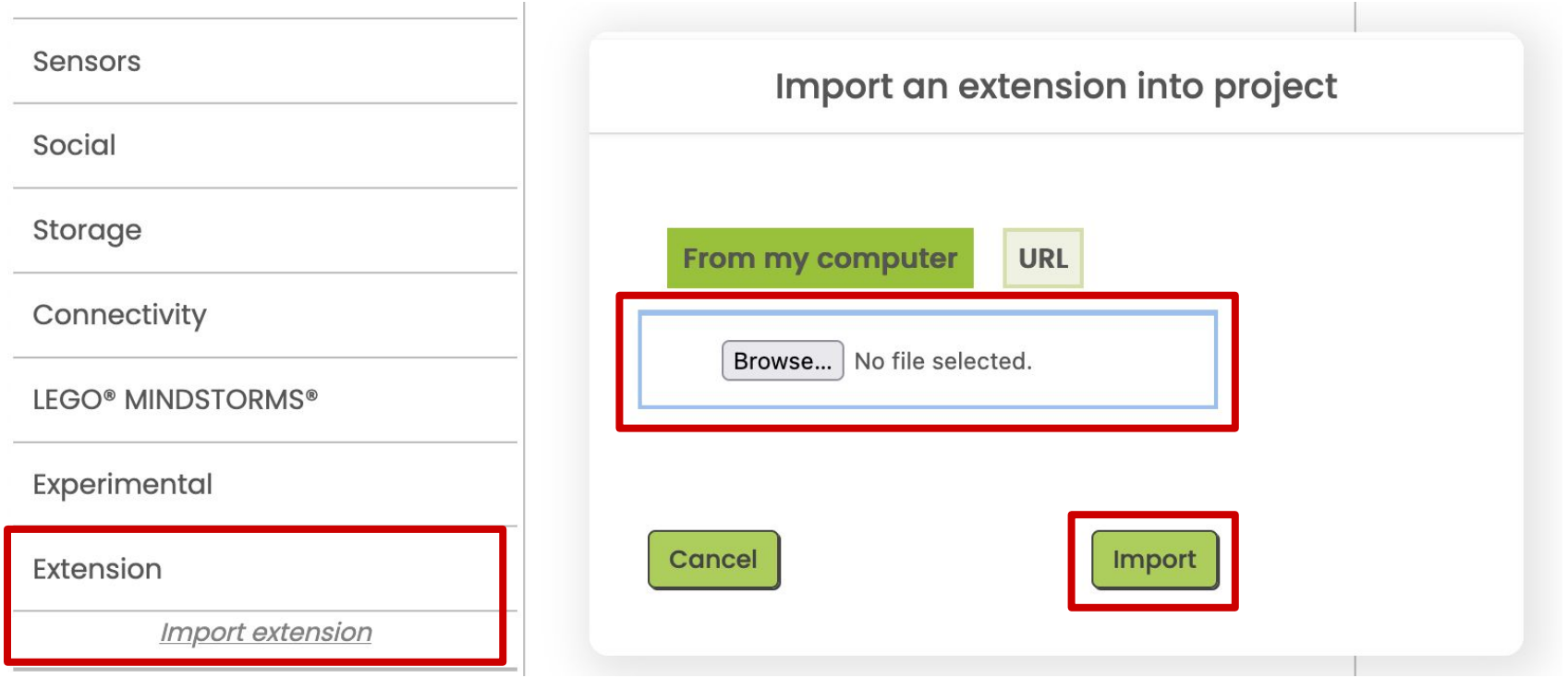
Extending App Inventor

```
private final BLEResponseHandler<String> writeHandler =
  new BLEResponseHandler<>() {
    @Override
    public void onWrite(String service, String characteristic,
                        List<String> values) {
      String message = "";
      if (values != null && values.size() > 0) {
        message = values.get(0);
      }
      MessageSent(message);
    }
  };
```

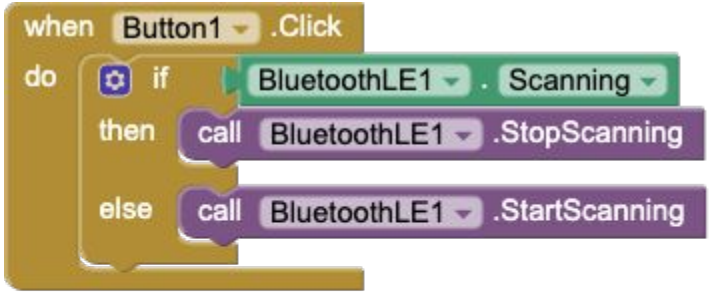
Extending App Inventor - Compiling

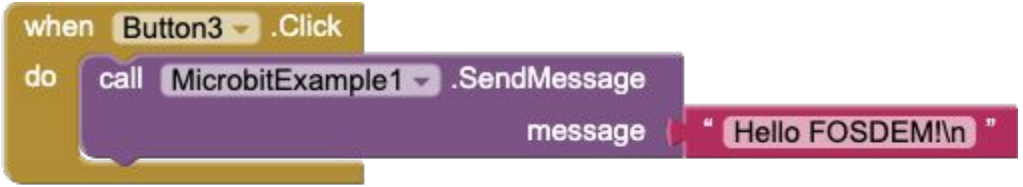
```
$ ant extensions
```

Testing


Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
Import extension
Import an extension into project
From my computer
URL
Browse...
No file selected.
Cancel
Import

Sample Blocks





Resources

Learn: appinventor.mit.edu

Build: ai2.appinventor.mit.edu

Contribute: https://github.com/mit-cml/appinventor-sources

Community: https://community.appinventor.mit.edu

Contact: ewpatton@mit.edu