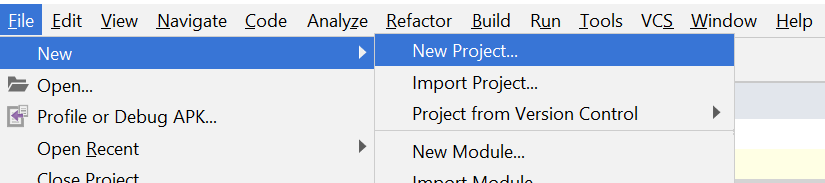
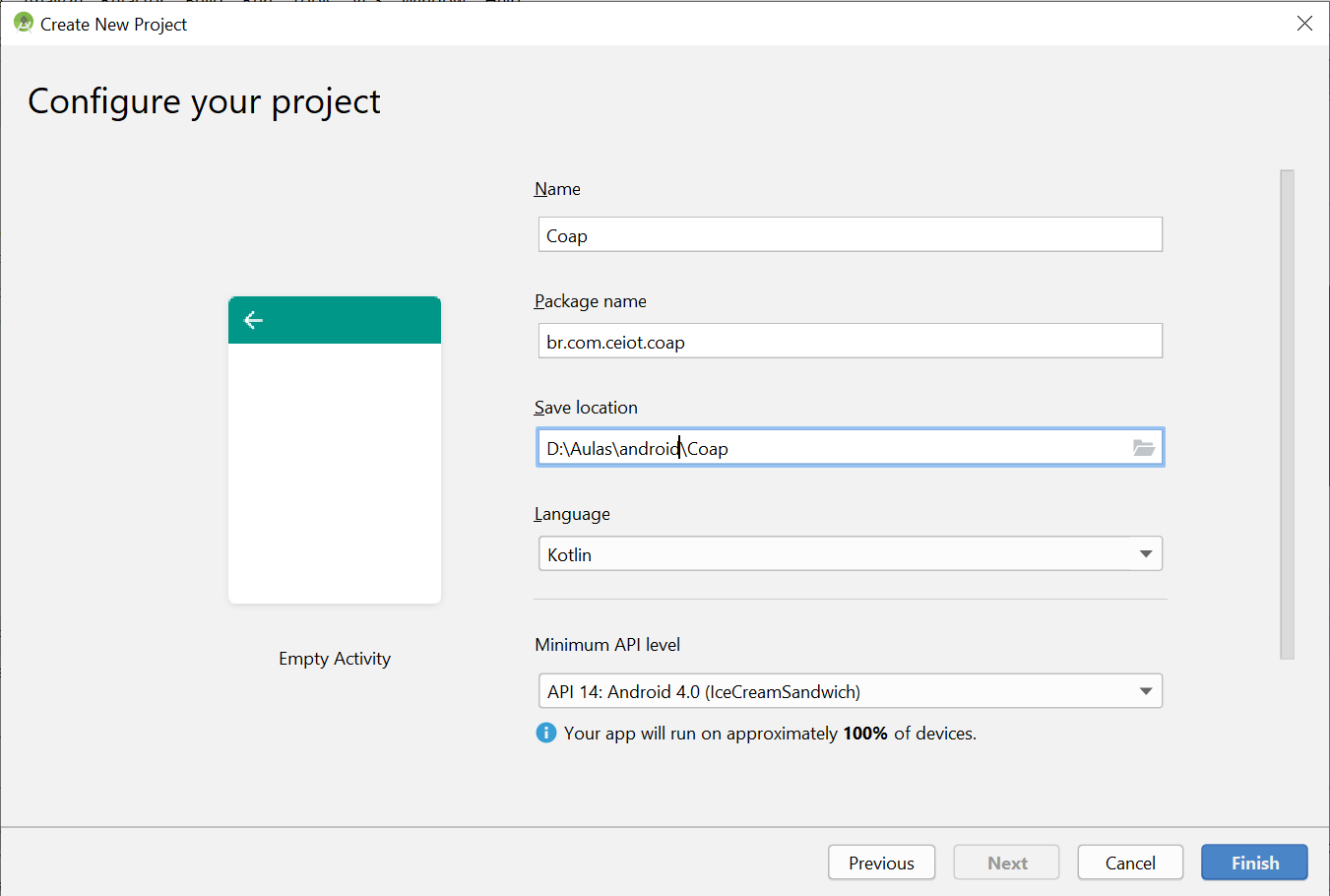
**Tutorial COAP Kotlin**

1. Criar um projeto no Android Studio:



1. Selecionar Empty Activity



1. Em build gradle (Project: coap), adicionar o repositório:

|  |  |
| --- | --- |
| 1  2  3 | maven {  url 'https://repo.eclipse.org/content/repositories/californium/'} |

1. Em build gradle (Project: app), adicionar as dependências:

|  |  |
| --- | --- |
| 1 | implementation 'org.eclipse.californium:californium-core:2.0.+' |

1. Em build gradle (Project: app), adicionar as seguintes opções:

|  |  |
| --- | --- |
|  | packagingOptions {  exclude **'3rd-party/APACHE-LICENSE-2.0.txt'** exclude **'epl-v10.html'** exclude **'edl-v10.html'** exclude **'3rd-party/cc0-legalcode.html'** exclude **'3rd-party/BSD-3-Clause-LICENSE.txt'** exclude **'3rd-party/MIT-license.html'** exclude **'notice.html'** } |

1. Registrar o serviço COAP no arquivo Manifest.xml:

|  |  |
| --- | --- |
| 1 | <**service android:name=".CaliforniumCoapServer"**/> |

1. Registrar permissões:

|  |  |
| --- | --- |
| 1  2  3  4 | <**uses-permission android:name="android.permission.WAKE\_LOCK"** /> <**uses-permission android:name="android.permission.INTERNET"** /> <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** /> <**uses-permission android:name="android.permission.READ\_PHONE\_STATE"** /> |

1. Adicionar classe AppExecutors. Esta classe servirá para gerenciar a execução de Threads fora e dentro da UI principal do aplicativo.

|  |  |
| --- | --- |
|  | **class** AppExecutors **private constructor**(  **private val diskIO**: Executor,  **private val networkIO**: Executor,  **private val mainThread**: Executor ) {   **fun** diskIO(): Executor {  **return diskIO** }   **fun** mainThread(): Executor {  **return mainThread** }   **fun** networkIO(): Executor {  **return networkIO** }   **private class** MainThreadExecutor : Executor {  **private val mainThreadHandler** = Handler(Looper.getMainLooper())   **override fun** execute(command: Runnable) {  **mainThreadHandler**.post(command)  }  }   **companion object** {   *// For Singleton instantiation* **private val LOCK** = Any()  **private var sInstance**: AppExecutors? = **null   val instance**: AppExecutors?  **get**() {  **if** (**sInstance** == **null**) {  *synchronized*(**LOCK**) **{  sInstance** = AppExecutors(  Executors.newSingleThreadExecutor(),  Executors.newFixedThreadPool(3),  MainThreadExecutor()  )  **}** }  **return sInstance** }  } } |

1. A classe CoapServerResource implementa um recurso para o COAP.

|  |  |
| --- | --- |
|  | **class** CoapServerResource(name: String) : CoapResource(name) {   **override fun** handleGET(exchange: CoapExchange) {  *// respond to the request* exchange.respond(**"Olá CEIOT"**)  } } |

1. Criar classe CaliforniumCoapServer. Ela é inicializada como um Serviço Android para implementação do Servidor COAP e registro de seus recursos.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **class** CaliforniumCoapServer : Service() {   **internal var server**: CoapServer? = **null   override fun** onCreate() {  **this**.**server** = CoapServer()  **server**?.add(CoapServerResource(**"CEIOTCoapServer"**))  }   **override fun** onStartCommand(intent: Intent, flags: Int, startId: Int): Int {  **server**?.start()  **return** Service.*START\_STICKY* }   **override fun** onDestroy() {  **server**?.destroy()  }   **override fun** onBind(intent: Intent): IBinder? {  **return null** } } |

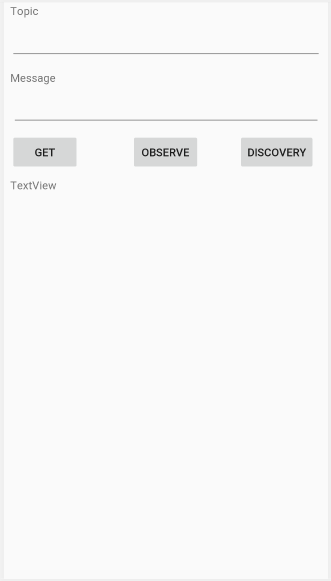
1. A classe CaliforniumSimpleClient implementa as funções GET, Observe e Discovery para um cliente COAP.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | **class** CaliforniumSimpleClient {   */\*\*  \* Implementação da função get  \** ***@param path*** *\** ***@return*** *\*/* **operator fun** get(path: String): String? {  **try** {  **val** client = CoapClient(**BASE\_URL** + path)  **val** response = client.get()  **if** (response != **null**) {  *println*(response.*code*)  *println*(response.*options*)  *println*(response.*payload*)  **return** response.*responseText* } **else** {  **return "No response received."** }  } **catch** (e: Exception) {  e.printStackTrace()  }   **return null** }   */\*\*  \* Implementação do observe  \** ***@param path*** *\** ***@param handler*** *\** ***@return*** *\*/* **fun** observe(path: String, handler: CoapHandler): String? {  **try** {  **val** client = CoapClient(**BASE\_URL** + path)  **val** relation = client.observe(handler)  **return if** (relation != **null** && relation.*current* != **null**) {  relation.*current*.*responseText* } **else** {  **"No response received."** }  } **catch** (e: Exception) {  e.printStackTrace()  }   **return null** }   */\*\*  \* Implementação do discovery  \** ***@param path*** *\** ***@return*** *\*/* **fun** discovery(path: String): String? {  **try** {  **val** client = CoapClient(**BASE\_URL** + path)  **val** set = client.discover()   **var** itens = **""  for** (wl **in** set) {  itens = itens + wl.*uri* + **"\r\n"** }  **return** itens  } **catch** (e: Exception) {  e.printStackTrace()  }   **return null** }   **companion object** {  **internal val BASE\_URL** = **"coap://localhost:5683/"***//"coap://californium.eclipse.org:5683/";* } } |

1. Na classe MainActivity, são realizados os vínculos com a interface do usuário e declarados os métodos que executam os métodos do cliente COAP.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89 | **package** br.com.ceiot.coap  **import** android.content.Intent **import** android.support.v7.app.AppCompatActivity **import** android.os.Bundle **import** android.view.View **import** kotlinx.android.synthetic.main.activity\_main.\* **import** org.eclipse.californium.core.CoapHandler **import** org.eclipse.californium.core.CoapResponse  **class** MainActivity : AppCompatActivity() {   *//****TODO (7) Criação das entiddades que serão utilizadas para o COAP* internal var californiumSimpleClient**: CaliforniumSimpleClient = CaliforniumSimpleClient();  **internal var mainCoapHandler**: MainCoapHandler = MainCoapHandler();   **override fun** onCreate(savedInstanceState: Bundle?) {  **super**.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   startService(Intent(**this**, CaliforniumCoapServer::**class**.*java*))  }   **override fun** onDestroy() {  stopService(Intent(**this**, CaliforniumCoapServer::**class**.*java*))  **super**.onDestroy()  }   */\*\*  \** ***TODO (8)*** *\* Método que irá implementar a ação do botão em tela para realizar um GET  \** ***@param view*** *\*/* **fun** getCoap(view: View) {  **val** topic = editText\_publish\_topic.getText().toString()  AppExecutors.**instance**!!.networkIO().execute(*Runnable* **{  var** result = **californiumSimpleClient**.get(topic)  AppExecutors.**instance**!!.mainThread().execute(*Runnable* **{** textView\_result.*text* = result  **}**)  **}**)  }   */\*\*  \** ***TODO (9)*** *\* Método que irá implementar a ação do botão em tela para realizar um discovery  \** ***@param view*** *\*/* **fun** discovery(view: View) {  **val** topic = editText\_publish\_topic.getText().toString()  *//CoapDiscoveryTask().execute(topic)* AppExecutors.**instance**!!.networkIO().execute(*Runnable* **{  var** result = **californiumSimpleClient**.discovery(topic)  AppExecutors.**instance**!!.mainThread().execute(*Runnable* **{** textView\_result.*text* = result  **}**)  **}**)  }   */\*\*  \** ***TODO (10)*** *\* Método que irá implementar a ação do botão em tela para realizar um Observer  \** ***@param view*** *\*/* **fun** observe(view: View) {  **val** topic = editText\_publish\_topic.getText().toString()  AppExecutors.**instance**!!.networkIO().execute(*Runnable* **{  californiumSimpleClient**.observe(topic, **mainCoapHandler**);  **}**)  }   */\*\*  \** ***TODO (11)*** *\* Tratamento da resposta do observer.  \*/* **internal inner class** MainCoapHandler : CoapHandler {   **override fun** onLoad(response: CoapResponse) {  textView\_result.*text* = response.*responseText* }   **override fun** onError() {  textView\_result.*text* = **"onError"** }  } } |

1. Em activity\_main.xml, criar o seguinte layout. Com o código abaixo:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119 | *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.constraint.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity"**>   <**EditText  android:id="@+id/editText\_publish\_topic"  android:layout\_width="0dp"  android:layout\_height="49dp"  android:layout\_marginEnd="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginRight="8dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="4dp"  android:ems="10"  android:inputType="textPersonName"  android:text=""  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView\_topic"**/>   <**EditText  android:id="@+id/editText\_publish\_msg"  android:layout\_width="392dp"  android:layout\_height="45dp"  android:layout\_marginTop="8dp"  android:ems="10"  android:inputType="textPersonName"  app:layout\_constraintTop\_toBottomOf="@+id/textView\_message"  android:layout\_marginEnd="8dp" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginRight="8dp"  android:layout\_marginStart="8dp" app:layout\_constraintStart\_toStartOf="parent"  android:layout\_marginLeft="8dp"**/>   <**Button  android:id="@+id/button\_get\_coap"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="8dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="8dp"  android:onClick="getCoap"  android:text="Get"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/editText\_publish\_msg"** />   <**Button  android:id="@+id/button\_observe"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"   android:layout\_marginEnd="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginRight="8dp"  android:layout\_marginStart="8dp"  android:onClick="observe"  android:text="Observe"  app:layout\_constraintEnd\_toStartOf="@+id/button\_dicovery\_coap"  app:layout\_constraintHorizontal\_bias="0.587"  app:layout\_constraintStart\_toEndOf="@+id/button\_get\_coap"  app:layout\_constraintTop\_toBottomOf="@+id/editText\_publish\_msg"  android:layout\_marginTop="8dp"**/>   <**Button  android:id="@+id/button\_dicovery\_coap"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginEnd="16dp"  android:layout\_marginRight="16dp"  android:layout\_marginTop="8dp"  android:onClick="discovery"  android:text="Discovery"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/editText\_publish\_msg"** />   <**TextView  android:id="@+id/textView\_topic"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginEnd="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginRight="8dp"  android:layout\_marginStart="8dp"  android:text="Topic"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"**/>   <**TextView  android:id="@+id/textView\_message"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginEnd="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginRight="8dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="12dp"  android:text="Message"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/editText\_publish\_topic"**/>   <**ScrollView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:layout\_marginBottom="8dp"  android:layout\_marginEnd="8dp"  android:layout\_marginLeft="8dp"  android:layout\_marginRight="8dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="8dp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/button\_observe"**>   <**TextView  android:id="@+id/textView\_result"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="TextView"  tools:layout\_editor\_absoluteX="163dp"  tools:layout\_editor\_absoluteY="433dp"** />  </**ScrollView**>  </**android.support.constraint.ConstraintLayout**> |