

# com.fr.stable.fun.EmailServiceProvider

- 
- 
- 
- 
- 
- 
- 
- 
- 

[EmailSendProvider](#)EmailServiceProvider

EmailServiceProvider[EmailSendProvider](#)

## EmailServiceProvider.java

```
package com.fr.stable.fun;

import com.fr.stable.email.EmailAttachmentProvider;
import com.fr.stable.fun.mark.Selectable;

import javax.mail.MessagingException;

/**
 * @author Cloud.Liu
 * @version 10.0
 * Created by Cloud.Liu on 2020/5/20
 */
public interface EmailServiceProvider extends Selectable {

    int CURRENT_LEVEL = 1;

    String XML_TAG = "EmailServiceProvider";

    /**
     *
     *
     * @param toAddress
     * @param ccAddress
     * @param bccAddress
     * @param fromAddress
     * @param subject
     * @param bodyContent
     * @param attaches
     * @param format
     * @param contentAttaches
     * @throws MessagingException
     */
    void send(String toAddress, String ccAddress, String bccAddress, String fromAddress, String subject, String
bodyContent, EmailAttachmentProvider[] attaches, String format, EmailAttachmentProvider[] contentAttaches)
throws MessagingException;
}
```

FR	10.0		
BI	5.1.3		

#### plugin.xml

```
<extra-core>  
  <EmailServiceProvider class="your class name"/>  
</extra-core>
```

```

package com.fr.base;

...

@EnableMetrics
public class EmailManager extends UniqueKey implements EmailManagerProvider, EmailServiceProvider {

    ...

    /**
     * {@link EmailServiceProvider}
     *
     * @param toAddress
     * @param ccAddress
     * @param bccAddress
     * @param fromAddress
     * @param subject
     * @param bodyContent
     * @param attaches
     * @param format
     * @param contentAttaches
     * @throws MessagingException
     * @date 2019-10-10
     */
    @Override
    public void send(String toAddress, String ccAddress, String bccAddress, String fromAddress, String subject,
String bodyContent, EmailAttachmentProvider[] attaches, String format, EmailAttachmentProvider[]
contentAttaches) throws MessagingException {
        Set<EmailServiceProvider> providers = getEmailServiceProvider();
        if (providers.isEmpty()) {
            sendEmail(toAddress, ccAddress, bccAddress, fromAddress, subject, bodyContent, attaches, format,
contentAttaches);
        } else {
            boolean sent = false;
            for (EmailServiceProvider provider : providers) {
                if (provider.selector().accept(ObjectHolder.wrap(toAddress))) {
                    provider.send(toAddress, ccAddress, bccAddress, fromAddress, subject, bodyContent,
attaches, format, contentAttaches);
                    sent = true;
                }
            }
            if (!sent) {
                FineLoggerFactory.getLogger().error("No email services accept this email: " + subject + " to "
+ toAddress + ".");
            }
        }
    }

    ...

    private Set<EmailServiceProvider> getEmailServiceProvider() {
        ExtraClassManagerProvider provider = PluginModule.getAgent(PluginModule.ExtraCore);
        if (provider != null) {
            return provider.getArray(EmailServiceProvider.XML_TAG);
        } else {
            return new HashSet<EmailServiceProvider>();
        }
    }

    ...
}

```

EmailServiceProviderEmailServiceProviderEmailSendProviderEmailSendProviderEmailServiceProvider

demodemo-email-service-provider

open-JSD-8153