{"version":3,"file":"config.js","sources":["../../src/util/config.js"],"sourcesContent":["/\*\*\n \* --------------------------------------------------------------------------\n \* Bootstrap (v5.2.0): util/config.js\n \* Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)\n \* --------------------------------------------------------------------------\n \*/\n\nimport { isElement, toType } from './index'\nimport Manipulator from '../dom/manipulator'\n\n/\*\*\n \* Class definition\n \*/\n\nclass Config {\n // Getters\n static get Default() {\n return {}\n }\n\n static get DefaultType() {\n return {}\n }\n\n static get NAME() {\n throw new Error('You have to implement the static method \"NAME\", for each component!')\n }\n\n \_getConfig(config) {\n config = this.\_mergeConfigObj(config)\n config = this.\_configAfterMerge(config)\n this.\_typeCheckConfig(config)\n return config\n }\n\n \_configAfterMerge(config) {\n return config\n }\n\n \_mergeConfigObj(config, element) {\n const jsonConfig = isElement(element) ? Manipulator.getDataAttribute(element, 'config') : {} // try to parse\n\n return {\n ...this.constructor.Default,\n ...(typeof jsonConfig === 'object' ? jsonConfig : {}),\n ...(isElement(element) ? Manipulator.getDataAttributes(element) : {}),\n ...(typeof config === 'object' ? config : {})\n }\n }\n\n \_typeCheckConfig(config, configTypes = this.constructor.DefaultType) {\n for (const property of Object.keys(configTypes)) {\n const expectedTypes = configTypes[property]\n const value = config[property]\n const valueType = isElement(value) ? 'element' : toType(value)\n\n if (!new RegExp(expectedTypes).test(valueType)) {\n throw new TypeError(\n `${this.constructor.NAME.toUpperCase()}: Option \"${property}\" provided type \"${valueType}\" but expected type \"${expectedTypes}\".`\n )\n }\n }\n }\n}\n\nexport default Config\n"],"names":["Config","Default","DefaultType","NAME","Error","\_getConfig","config","\_mergeConfigObj","\_configAfterMerge","\_typeCheckConfig","element","jsonConfig","isElement","Manipulator","getDataAttribute","constructor","getDataAttributes","configTypes","property","Object","keys","expectedTypes","value","valueType","toType","RegExp","test","TypeError","toUpperCase"],"mappings":";;;;;;;;;;;;;;;EAAA;EACA;EACA;EACA;EACA;EACA;EAKA;EACA;EACA;;EAEA,MAAMA,MAAN,CAAa;EACX;EACkB,EAAA,WAAPC,OAAO,GAAG;EACnB,IAAA,OAAO,EAAP,CAAA;EACD,GAAA;;EAEqB,EAAA,WAAXC,WAAW,GAAG;EACvB,IAAA,OAAO,EAAP,CAAA;EACD,GAAA;;EAEc,EAAA,WAAJC,IAAI,GAAG;EAChB,IAAA,MAAM,IAAIC,KAAJ,CAAU,qEAAV,CAAN,CAAA;EACD,GAAA;;IAEDC,UAAU,CAACC,MAAD,EAAS;EACjBA,IAAAA,MAAM,GAAG,IAAA,CAAKC,eAAL,CAAqBD,MAArB,CAAT,CAAA;EACAA,IAAAA,MAAM,GAAG,IAAA,CAAKE,iBAAL,CAAuBF,MAAvB,CAAT,CAAA;;MACA,IAAKG,CAAAA,gBAAL,CAAsBH,MAAtB,CAAA,CAAA;;EACA,IAAA,OAAOA,MAAP,CAAA;EACD,GAAA;;IAEDE,iBAAiB,CAACF,MAAD,EAAS;EACxB,IAAA,OAAOA,MAAP,CAAA;EACD,GAAA;;EAEDC,EAAAA,eAAe,CAACD,MAAD,EAASI,OAAT,EAAkB;EAC/B,IAAA,MAAMC,UAAU,GAAGC,eAAS,CAACF,OAAD,CAAT,GAAqBG,4BAAW,CAACC,gBAAZ,CAA6BJ,OAA7B,EAAsC,QAAtC,CAArB,GAAuE,EAA1F,CAD+B;;EAG/B,IAAA,OAAO,EACL,GAAG,IAAKK,CAAAA,WAAL,CAAiBd,OADf;QAEL,IAAI,OAAOU,UAAP,KAAsB,QAAtB,GAAiCA,UAAjC,GAA8C,EAAlD,CAFK;EAGL,MAAA,IAAIC,eAAS,CAACF,OAAD,CAAT,GAAqBG,4BAAW,CAACG,iBAAZ,CAA8BN,OAA9B,CAArB,GAA8D,EAAlE,CAHK;EAIL,MAAA,IAAI,OAAOJ,MAAP,KAAkB,QAAlB,GAA6BA,MAA7B,GAAsC,EAA1C,CAAA;OAJF,CAAA;EAMD,GAAA;;IAEDG,gBAAgB,CAACH,MAAD,EAASW,WAAW,GAAG,IAAKF,CAAAA,WAAL,CAAiBb,WAAxC,EAAqD;MACnE,KAAK,MAAMgB,QAAX,IAAuBC,MAAM,CAACC,IAAP,CAAYH,WAAZ,CAAvB,EAAiD;EAC/C,MAAA,MAAMI,aAAa,GAAGJ,WAAW,CAACC,QAAD,CAAjC,CAAA;EACA,MAAA,MAAMI,KAAK,GAAGhB,MAAM,CAACY,QAAD,CAApB,CAAA;EACA,MAAA,MAAMK,SAAS,GAAGX,eAAS,CAACU,KAAD,CAAT,GAAmB,SAAnB,GAA+BE,YAAM,CAACF,KAAD,CAAvD,CAAA;;QAEA,IAAI,CAAC,IAAIG,MAAJ,CAAWJ,aAAX,EAA0BK,IAA1B,CAA+BH,SAA/B,CAAL,EAAgD;EAC9C,QAAA,MAAM,IAAII,SAAJ,CACH,GAAE,IAAKZ,CAAAA,WAAL,CAAiBZ,IAAjB,CAAsByB,WAAtB,EAAoC,aAAYV,QAAS,CAAA,iBAAA,EAAmBK,SAAU,CAAuBF,qBAAAA,EAAAA,aAAc,IAD1H,CAAN,CAAA;EAGD,OAAA;EACF,KAAA;EACF,GAAA;;EAhDU;;;;;;;;"}